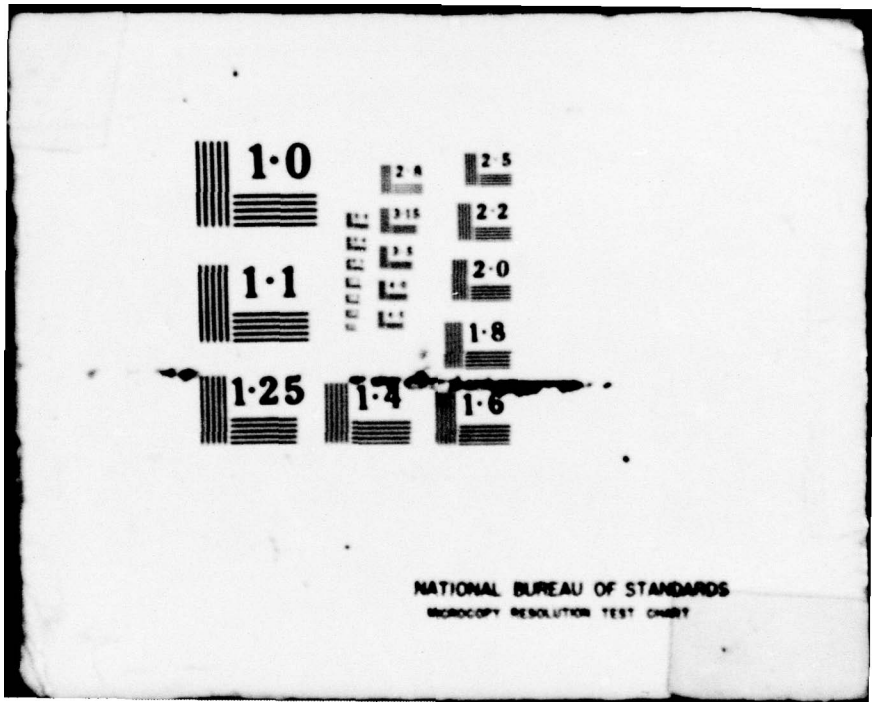
AD-A075 131 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 17/1
PHASE INTEGRAL ALGORITHMS FOR SIGNAL SPEEDS IN RANGE-DEPENDENT --ETC(U)
JUL 78 R C SHOCKLEY
UNCLASSIFIED NOSC/TR-412 NL

1 OF 1
AD-
A075131

END
DATE
FILMED
11-79
DDC


1·0
1·1
1·25
2·5
2·2
2·0
1·8
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

NOSC

LEVEL II

NOSC TR 412

NOSC TR 412

Technical Report 412

PHASE INTEGRAL ALGORITHMS FOR SIGNAL SPEEDS IN RANGE-DEPENDENT SOUND VELOCITY PROFILES

R. C. Shockley

10 July 1978

Final Report: January 1978 — July 1978

Prepared for
Naval Electronic Systems Command
(PME 124)

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

RR GAVAZZI, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

This work pertains to a method for estimating from archival sound speed profile data the effective signal speed in range-dependent environments. The method employs the adiabatic invariance of the phase integral for underwater sound rays, and offers certain computational advantages in the context of improved time-difference-of-arrival performance using Multi-Array Correlation (MAC) equipment. The work was done under NOSC 24311N, X0522114, and was sponsored by Naval Electronic Systems Command (PME-124).

Released by
M. R. Akers, Head
System Concepts and Analysis Division

Under Authority of
E. B. Tunstall, Head
Ocean Surveillance Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report 412 (TR 412)
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: NOSC/TR-412

4. TITLE (and Subtitle): PHASE INTEGRAL ALGORITHMS FOR SIGNAL SPEEDS IN RANGE-DEPENDENT SOUND VELOCITY PROFILES
5. TYPE OF REPORT & PERIOD COVERED: Final Report, Jan[illegible]–Jul[illegible]78
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): R. C. Shockley
8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Ocean Systems Center, San Diego, CA 92110
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 24311N, XO522114

11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Electronic Systems Command, Washington, DC 20360
12. REPORT DATE: 1[illegible] July [illegible]978
13. NUMBER OF PAGES: 47

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adiabatic invariant
Guided propagation
Range dependent profiles
SOFAR
Sound speed algorithms
Underwater sound

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The adiabatic invariance of the phase integral $J = \oint \sin\theta \, dz/c$ for ducted sound rays in SOFAR channels is demonstrated by means of exact raytracing solutions in range-dependent channels. FORTRAN algorithms for this and for relating J to the average ray speed in arbitrary ducts are presented and discussed in the context of acoustic surveillance.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 159

FOREWORD

This report presents two Fortran raytracing codes and related theory which are designed to treat underwater sound propagation in the presence of range-dependent sound velocity profiles (SVPs). Particular emphasis is placed on the average horizontal ray velocity and its variation as a ray traverses different SVPs. Efficient calculation of this variation is of great importance for improved modern multisensor localization techniques. The following pages discuss the adiabatic invariant approximation (AIA) and the use of the codes RAY1 and RAY2 in sufficient depth that the reader may apply them to his own problems or modify them for this purpose. This work was supported by the Naval Electronic Systems Command PME-124 under the Performance Evaluation and Prediction program at the Naval Ocean Systems Center. Portions of this work have appeared in the Journal of the Acoustical Society of America.

OBJECTIVE

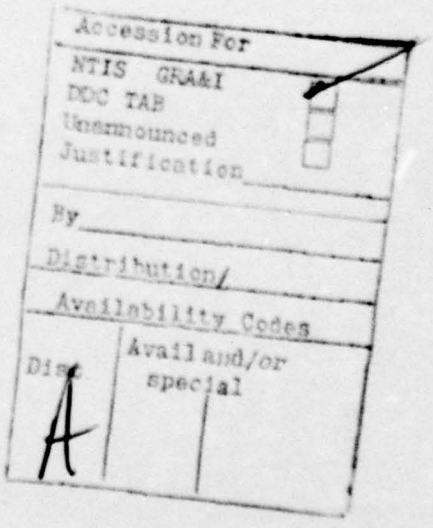
The objective of the work was to calculate efficiently the variation of the average horizontal ray velocity as the ray traverses different sound velocity profiles. Efficient calculation of this variation is of great importance for improved modern multisensor localization techniques.

RESULTS

The adiabatic invariant approximation agrees well with exact raytracing results even for strongly inhomogeneous sound velocity profiles. Simple algorithms are developed for demonstrating this and for computing the adiabatic invariant in arbitrary profiles for a user-selected family of rays.

RECOMMENDATIONS

A set of curves or tables should be prepared showing the functional relation between the adiabatic invariant, the average sound speed, and the ray angle, for areas of high interest in acoustic surveillance. These would provide a data base for improved multisensor localization and would reduce the calculation of average ray speeds to a table-look-up procedure once the location of the source is known approximately.



CONTENTS

I. INTRODUCTION

Several important problems in underwater sound, ionospheric propagation, and seismic research involve the presence of ducts or waveguides which effectively trap the energy, greatly reducing transmission loss over the corresponding case of an isotropic medium. Often, however, the duct varies with range, precluding exact normal mode solutions and affecting signal speeds in a generally complicated fashion. The range variation may consist of changes in the physical dimensions of the duct or in the index of refraction across the duct.

Simple computational methods are desired for estimating the influence of such range-dependence. The requirements basically are that the procedure be readily checked against known exact solutions, that it be relatively inexpensive to implement using archival data bases, and that revisions of the data should be readily incorporated in the method, without requiring massive recomputation.

An attractive candidate for meeting these needs is the adiabatic invariant approximation (AIA). This presupposes that in the cases of interest any range-dependent variations take place over distances large compared to a single ray cycle. In other words, the properties of the medium should be sensibly constant over a cycle. Under such conditions, which nearly always apply for deep sound channels, it turns out that ray paths evolve in range so as to keep a certain quantity (J) constant, known as the action integral or phase integral.[1] This "conservation law" allows one to follow a ray, in essence, without ever tracing it over its entire path. It is only necessary to know in each local region how the quantities of interest (eg, maximum depth of the ray, vertexing speed, average horizontal velocity, etc.) are related to the adiabatic invariant J, something easily done simply by tracing a few ray cycles in each local region.

Although we have been speaking in terms of rays, adiabatic invariants appear in mode theory as well. Each mode corresponds to a particular vertexing depth and speed for the associated family of rays. In referring only to rays, as we shall do throughout what follows, we are assuming the duct supports so many modes that they comprise a continuum in effect; their discrete nature may be disregarded without significant loss of accuracy. Since deep sound channels are of the order of ten or more wavelengths across at frequencies of interest for time difference fixing, the mode-continuum approximation is generally quite good. The low order modes are not strongly excited by acoustic sources, which are usually far from the sound channel axis.

When one has only low order modes, the adiabatic invariants are the mode numbers themselves. This is intuitively plausible, for if one considers an arbitrarily gradual evolution of one duct into a second one, having, say, a somewhat different index profile, then it is clear that the power initially in a particular mode must equal the power in that mode in the altered duct. Only rather strong perturbations will cause mode coupling; in their absence, there is no cause for energy to distribute itself among any other discrete modes.

---

1. Weston, DE, *Guided Propagation in a Slowly Varying Medium*, Proc. Phys. Soc. (London) v. 73, p. 365-384, 1958.

Exceptions to this simple picture obviously exist in cases where one duct evolves into two. In such cases one cannot treat the splitting of acoustic energy accurately unless the details of the SVP evolution are considered. Fortunately, such cases are rare. The procedures described in this report should be applicable to all ocean areas of surveillance interest.

The computer codes described here serve two purposes. RAY1 allows one to set up a radically inhomogeneous medium whose SVP has a prescribed functional form and to determine numerically by how much the approximation J = constant fails for various horizontal gradients. RAY2 allows one to input an arbitrary but range-independent SVP and to trace a family of rays in order to find the relation between J and other properties of interest for rays in that particular SVP. The former code is of primarily theoretical interest, and serves to establish the accuracy for the AIA in analytically tractable cases, while it is our hope that the latter will serve as a model for implementation of AIA methods in acoustic surveillance.

In section II we describe in general terms the scope of RAY1 and RAY2, we review briefly the adiabatic invariant theory as applied to acoustic rays, the analogous problem in classical mechanics, and finally give the AIA signal speed algorithm. Sections III and IV provide instructions in detail on the procedure for executing RAY1 and RAY2. In section V, we shall treat the internal logical structure of RAY1 and RAY2 in order to pave the way for any changes deemed necessary or desirable. Listings of the Fortran source code appear in section VI.

II. GENERAL DESCRIPTION OF RAY1 AND RAY2

In this section we shall describe the general features of RAY1 and RAY2 and the theory behind these codes. In sections III and IV we shall present specific information needed to make runs with these codes. Here we are concerned with the practical need for such codes, and the underlying concept of using an adiabatic invariant as a means for following the evolution of rays in range-dependent SVPs.

A. THE ADIABATIC INVARIANT FOR SOUND RAYS

The basic problem of time-difference fixing provides a context for introducing the adiabatic invariant concept. For simplicity, suppose that acoustic signals propagate with the same speed in the ocean regardless of direction. In this isotropic ocean, a pair of sensors which receive a given signal may determine, based on the difference of arrival times, a locus of points on which the source must lie. The locus is a hyperbola with the sensors as foci. If R_1 and R_2 represent distances to sensors 1 and 2 respectively, and $c\Delta t$ represents the equivalent delay length, then $|R_1 - R_2| = c\Delta t$ determines the hyperbola. A third sensor, or some additional data from one of the original two, then suffices to locate the source uniquely. (Ambiguities arising in particular geometries are not of interest here, although they may be important in practice.)

The procedure just described works only approximately in actual ocean environments. For long-range detections on the order of several hundred kilometers the signal speed generally will vary with range. Modern acoustic surveillance systems are capable of sufficiently precise measurement that this inhomogeneity poses a major limitation on localization. How can one determine an appropriate average sound speed so as to construct the needed locus?

A natural solution is to use a conserved quantity for sound rays in range-dependent environments. Weston[1] has pointed out that the quantity

$$J = \oint c^{-1} \sin\theta \, dz \tag{1}$$

is conserved provided that changes in the SVP are sufficiently gradual, or "adiabatic." Here c is the local depth-dependent sound speed, and is regarded as being range-independent over the period of integration (one ray cycle), θ is the ray angle measured with respect to horizontal, and z is the depth coordinate (usually taken as positive downwards). More detailed discussion of the invariance of J in range-dependent SVPs, as well as a short bibliography on the topic, will be found in reference 2. A proof of the adiabatic invariance of the action integral will be found in reference 3. At this point it may be of interest to consider the mechanical analogue of trapped sound rays and eq (1).

1. Classical Mechanical Analogy

Ray theory is of course an approximation to an "exact" wave theory of propagation. In the same sense, classical mechanics is an approximation to the more precise quantum mechanical theory. In both cases the smallness of the wavelength with respect to the physical dimensions of the medium or measuring apparatus involved permits one to pass over to the geometrical optics limit, ignoring small wavelength-dependent corrections. In classical mechanics, the motion of a system is obtainable from Hamilton's principle, which is a variational principle, stating that the actual trajectory followed is such as to minimize the integral $\int_{t_1}^{t_2} L(\dot{q}, q\ ; t)\, dt$. (Actually the statement of Hamilton's principle permits maximization as well, but we shall not be concerned with this in what follows.) Here L is the Lagrangian, a function of the coordinates and velocities, symbolically denoted q and $\dot{q}$, respectively, and possibly of time t. The coordinates and velocities are functions of time. The system moves from some fixed starting point $(q(t_1), \dot{q}(t_1))$ to a final point $(q(t_2), \dot{q}(t_2))$, also fixed. Hamilton's principle is written

$$\delta \int_{t_1}^{t_2} L\, dt = 0\ , \tag{2}$$

where δ indicates an infinitesimal variation of the path. Equation (2) states that the system follows a minimizing trajectory, since any infinitesimal variation about the trajectory must produce no change in the value of the integral.

In classical mechanics, the equations of motion follow if one uses for the Lagrangian L = T - V, where T is the kinetic energy and V the potential energy of the system. Then the calculus of variations yields, as a theorem from eq (2), Newton's equation of motion.

---

2. Shockley, RC, *Paraxial and Nonparaxial Ray Speeds in Strongly Range-Dependent SOFAR Channels*, J. Acoust. Soc. Am. v. 64, p. 1171-1177 (1978).

3. Landau, LD and Lifshitz, EM, *Mechanics*, Pergamon Press, Reading, Mass., p. 154-158, 1960.

Variational principles (with different choices for L) also lead to the equations of electrodynamics, hydrodynamics, quantum mechanics, etc., in similar fashion.

If we consider Fermat's principle, the similarity to Hamilton's principle is clear. Fermat's principle states that a ray will follow that trajectory minimizing the time of flight between a given initial and final point:

$$\delta \int_1^2 dt = \delta \int_1^2 ds/c = 0 \tag{3}$$

where ds is an infinitesimal element of length along the ray path and c the speed of propagation in the medium.

For concreteness, we restrict our discussion to a slab symmetric medium and let x be the horizontal and z the vertical coordinate. Then eq (3) becomes

$$\delta \int_1^2 c^{-1} (1 + \dot{z}^2)^{1/2} dx = 0 \tag{4}$$

where $\dot{z} \equiv dx/dx$ is the slope of the ray. This result now is identical to eq (2), Hamilton's principle, under the transformation

$$t \rightarrow x \ ,$$

$$q \rightarrow z \ ,$$

$$L(q, \dot{q}; t) \rightarrow (1 + \dot{z}^2)^{1/2}/c(z; x) \ . \tag{5}$$

Thus range plays the role of time in the mechanical problem, and the depth z is analogous to particle displacement. Hence $\dot{z} = \tan\theta$ is analogous to velocity. Note that L does not resemble a quantity like T-V.

It is natural to ask what equation involving the "acceleration" $\ddot{z}$ is analogous to Newton's equation of motion. The answer provides insight into the behavior of rays in deep sound channels and bears directly on the raytracing routine employed in RAY1 and RAY2. The Euler-Lagrange equation, which follows as a theorem[3] from the variational principle in eq (2), as mentioned above, states that

$$(d/dt)\, \partial L/\partial \dot{q} - \partial L/\partial q = 0 \ . \tag{6}$$

If the reader applies the transformation given in eq (5) then, after some algebraic manipulation, eq (6) will be seen to become

$$\ddot{z} = -(1 + \dot{z}^2)\, c^{-1}\, \partial c/\partial z + \dot{z}(1 + \dot{z}^2)\, c^{-1}\, \partial c/\partial x \ , \tag{7}$$

a result obtained previously by Milder.[2,4] Equation (7) is similar to Newton's equation of motion, giving the acceleration as a function of displacement, velocity, and time.

RAY1 and RAY2 use a fourth order Runge-Kutta scheme[2] to integrate eq (7), and thus obtain the ray depth as a function of range. Equation (7) describes the ray

---

4. Milder DM, *Ray and Wave Invariants for SOFAR Channel Propagation*, J. Acoust. Soc. Am., v. 46, p. 1259-1263 (1969).

curvature in terms of the local sound speed derivatives in depth and range and the ray slope. For prescribed initial conditions $z(0)$ and $\dot{z}(0)$, and a given function $c(z;x)$, it specifies a unique trajectory satisfying Fermat's principle.

When the medium is range-independent, the second term on the right-hand side of eq (7) vanishes. A further simplification comes about since in range-independent media Snell's law holds in the form $c_v = c/\cos\theta$, where c_v is the constant vertexing speed (where $\dot{z} = 0$). Thus eq (7) becomes

$$\ddot{z} = -c_v^2(dc/dz)\,c^{-3} = -(\partial/\partial z)(-c_v^2/2c^2) \quad . \tag{8}$$

Hence in range-independent SVPs the motion of a ray is precisely analogous to the trajectory of a mechanical particle of unit mass under the action of a potential field given by

$$V(z) = -c_v^2/2c^2 \quad . \tag{9}$$

This result allows considerable insight into the trajectories of rays. For example, if we consider the Hirsch profile

$$c(z) = c_0/(1 - z^2/a^2)^{1/2} \quad ,$$

then the potential is

$$V(z) = -(c_v/c_0)^2\,(1 - z^2/a^2)$$

$$= (c_v/c_0\,a)^2\,z^2 + \text{const} \quad ,$$

which is a quadratic function of z, the same as a simple harmonic oscillator. Ray paths must therefore be sine waves in this SVP. It is simple to sketch potential functions for any reasonable SVP, and range-dependence may be included in this analogy by allowing $V(z)$ to vary slowly with time.

When the medium is range-dependent, then the analogous classical mechanics problem involves a time-dependent potential, as one can see from the transformation eq (5). Hence the energy of the analogous particle is not conserved, nor are the positions of the turning points where $\dot{q} = 0$ or $\dot{z} = 0$. If the variations are gradual, however, the action integral

$$J = \oint p\,dq \tag{10}$$

is approximately conserved.[3] Here $p \equiv \partial L/\partial\dot{q}$; p is called the momentum canonically conjugate to the coordinate q in classical mechanics. If the reader again applies eq (5) and uses $\dot{z} = \tan\theta$, he will find

$$p = \dot{z}/c\,(1 + \dot{z}^2)^{1/2} = \sin\theta/c \quad ,$$

and thus eq (1) follows as the adiabatic invariant for sound rays. Note that the integral covers one cycle of the trajectory and represents the area of an (almost) closed loop traced

out in p-q space, known as phase space. The path is a closed loop for range-independent SVPs.

2. AIA Algorithm for Average Horizontal Ray Velocity

One may use the invariance of J to compute the average horizontal velocity of a ray traversing a range-dependent SVP as follows. We assume it is possible to break up the entire region of interest into local regions bounded by some set of contiguous contours inside each of which c(z) is a known function, which can be regarded as range-independent within the local region.

First one constructs tables showing the relationship between J and $\bar{c}$, the average horizontal ray velocity, for each local region. This may require tracing perhaps 10 to 50 rays for each SVP. Each ray need only be traced for a small number of cycles (one or two) to find J and $\bar{c}$ for the given initial conditions. These tables then allow one to find how $\bar{c}$ varies from region to region for a particular J value, by employing simple table-look-up procedures.

This method does not yield detailed information on the trajectory, such as positions of ray turning points in range, but this is the cost of obtaining estimates of $\bar{c}$ for long-range paths with simple algorithms. For a prescribed path, one then may find the effective signal speed for given initial conditions (depth and ray angle) from the $\bar{c}$ values in each region according to the fraction of time spent in each region. If R denotes the total range, R_i the distance traveled in region i, and $\bar{c}_i$ the average horizontal velocity in region i, then the effective signal speed is given by

$$\bar{c} = R/\sum_i (R_i/\bar{c}_i) \tag{11}$$

Additional algorithms are necessary to select a range of initial conditions that characterize rays which carry most of the acoustic energy. Depending on the geometry, these may range from simple inspection procedures to complicated ray-trace routines requiring detailed bathymetry charts. Reference 2 discusses possible methods for selecting appropriate ranges of J values using the output of RAY2 (see section VI). Further remarks on this are included in section IIC, where we indicate how certain of the quantities computed by RAY2 facilitate ray selection.

B. GENERAL DESCRIPTION OF RAY1

A listing of the code RAY1 will be found in section V, and instructions on its use in section III. Here we shall describe the general features of RAY1.

RAY1 provides quantitative answers to the question of the validity of the AIA. Earlier we noted that J is conserved for sufficiently slow variations in the SVP. More precisely, J is conserved (or nearly so) provided that any changes in the SVP are fractionally small over one cycle distance (in range) of the ray. This follows from the fundamental law of ray theory, Fermat's principle of least time[3] as discussed in section II.A.1. We refer the reader desiring further discussion of this to Landau and Lifshitz[3] and the bibliography of reference 2.

In real ocean environments, the adiabatic condition is satisfied for RR (Refracted Refracted) rays, but one still may ask what degree of inhomogeneity is required before the AIA fails badly. RAY1 provides specific answers to this by allowing the user to control the horizontal gradients of parameters in the SVP function. Thus both the degree of inhomogeneity and the effect of the functional form c(z) may be investigated.

Presently three self-similar SVPs are available as subroutines in RAY1. They are the parabolic profile,

$$c(z, x) = c_0(z) [1 + z^2/a^2 (x)] \quad , \tag{12a}$$

the Hirsch profile,

$$c(z, x) = c_0(z) [1 - z^2/a^2 (x)]^{-\frac{1}{2}} \quad , \tag{12b}$$

and the bilinear profile,

$$c(z, x) = c_0(x) [1 \pm z/a(x)] \quad , \qquad z > 0 \, (< 0) \tag{12c}$$

Figure 1 shows these SVPs for range-independent SVPs. Each of equations (3) describes a slab symmetric (depending only on range x and depth z) sound channel with an axis at $z = 0$, and an axial sound speed $c_0(x)$. Both $c_0(x)$ and $a(x)$ are linear functions of x in the current version of RAY1. By varying their derivatives with appropriate input data, the user may study the degree to which nonadiabatic changes in the SVP affect the action integral, or the discrepancy between the computed average horizontal ray velocity and the predicted value, assuming $\partial J/\partial x = 0$. Details on the program inputs will be deferred until section III.

RAY1 allows the user to select one of the three SVPs, and control tracing of rays in the presence of constricting (if $da/dx < 0$) or widening ($da/dx > 0$) sound channels. Gradients in c_0 and a are independent. In addition, the raytracing routine employed (a fourth order Runge-Kutta algorithm applied to the Euler-Lagrange equation of motion obtained from Fermat's principle) has an automatic test and redefinition routine which refines the step size dx when there are fewer than 150 steps per half-cycle, although this number can be easily modified (see section V).

For completeness, we present in table 1 the analytic formulae for ray paths and the average horizontal ray velocity in each of the SVPs in RAY1 for range-independent conditions. For the parabolic SVP, u_1 and $E(u_1)$ are the normal elliptic integrals of the first and second kind, respectively, with argument z/a, modulus squared $(\alpha-1)^2/2(\alpha^2+1)$ where $\alpha^2 \equiv c_v/c_0$; sn u_1, cn u_1 and dn u_1 are Jacobian elliptic functions; E(k), K(k) and $\Pi(\beta^2, k)$ are complete elliptic integrals of the first, second, and third kind with $k^2 = (\alpha-1)^2/(2\alpha^2+2)$ and the parameter $\beta^2 = (1-\cos\theta_0)/2$.[5] θ_0 is the angle with respect to horizontal at which the ray crosses the sound channel axis at $z = 0$ in all formulae.

Figures 2 and 3 show the agreement found for $\theta_0(0) = 30°$ in an earlier study[2] between predicted sound speeds, assuming $\partial J/\partial x = 0$, and those computed by RAY1 in the

5. Byrd, PF and Friedman, MD, *Handbook of Elliptic Integrals for Engineers and Scientists*, Springer Verlag, New York (1971).

presence of various gradients in a(x), the depth scaling function. The agreement generally is good even when the environment changes rapidly, for example by 40% (in a(x)) in a single ray cycle. Such agreement, although it cannot be expected for all SVP choices, lends considerable support to the general validity of the AIA. Quantitative answers to this question are of interest since the AIA method seems to offer a simple solution to some time difference fixing (TDF) problems: data bases for implementing AIA sound speed algorithms are easily generated, the table-look-up AIA computations typically are far less time-consuming than long-range raytracing, and the method permits very flexible updating of data in individual SVP regions without requiring long-range raytracing computations to be repeated.

In figures 2 and 3 x_p is the cycle distance and x_c the critical range at which the channel constricts to an extent such that rays are reflected, and cannot propagate further. The notation $x_c/x_p\,(x=0)$ is the ratio of the critical range to a cycle distance, evaluated at the initial range. Thus, for example, if $x_c/x_p = 3$, the critical constriction occurs only 3 cycle distances from the ray's starting point. The vertical axis is the normalized average horizontal ray speed ($\bar{c}^* = 1$ for a ray on the channel axis) and the horizontal axis is the normalized channel width parameter ($a^* = a/a_0$ where a is the scale length in the table 1 formulas, and a_0 is the initial value of a).

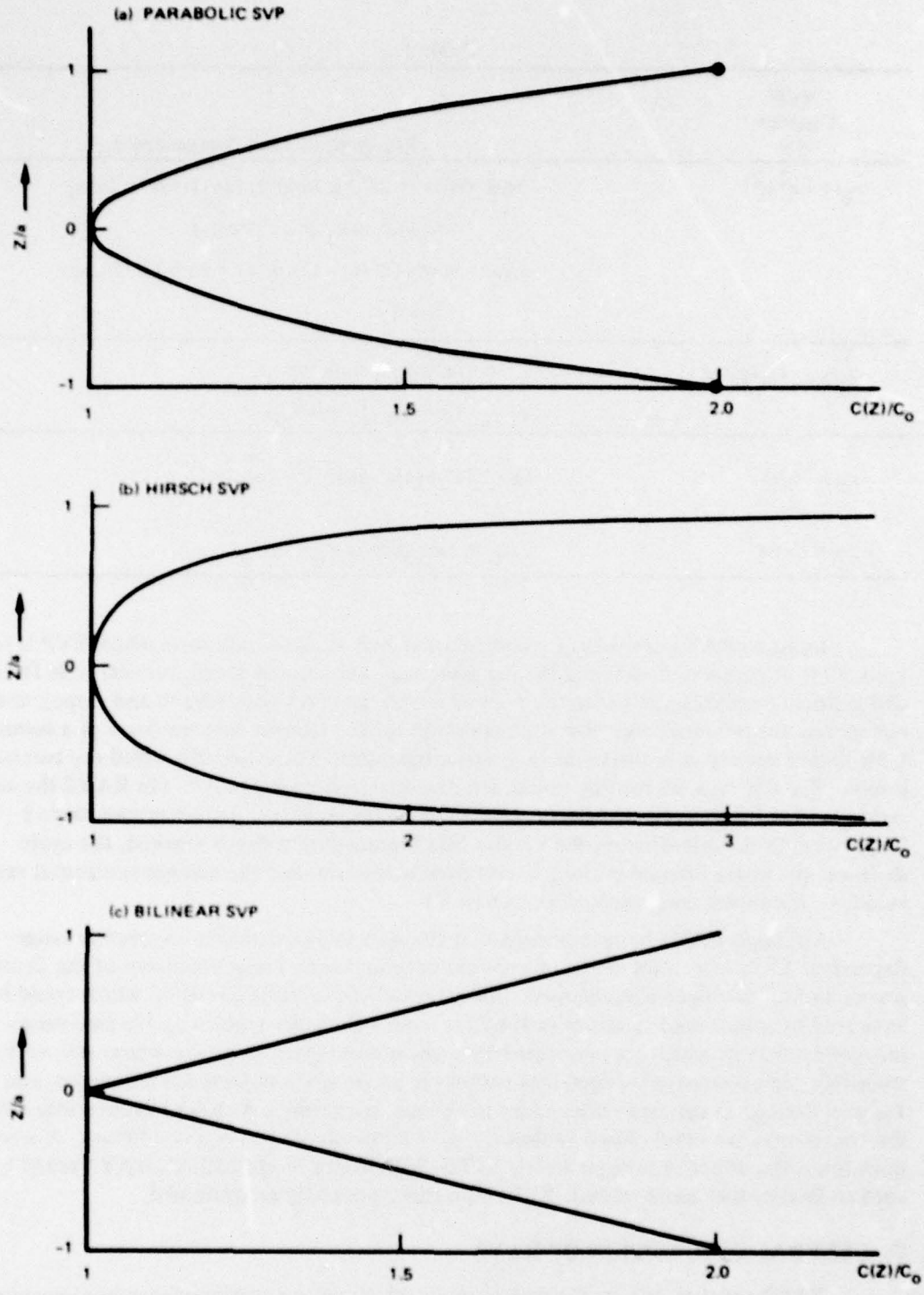


Figure 1. Three analytical forms for model sound velocity profiles (from eq (12)) in dimensionless form.

Table 1.

SVP Function c(z)	Ray Trajectory and Average Speed
$c_0(1 + z^2/a^2)$	$x/a = (2\alpha^2 + 2)^{-1/2}\,[(2\alpha^2 + 2)\,E(u_1) - (\alpha^2 + 2\alpha)u_1 - (\alpha - 1)^2\,sn\,u_1\,cn\,u_1 / dn\,u_1]$ $\bar{c}/c_0 = \sec\theta_0\,(2E(k) - K(k))/[(1 + \sec\theta_0)\,\Pi(\beta^2, k) - K(k)]$
$c_0(1 - z^2/a^2)^{-1/2}$	$z = a\sin\theta_0 \cos(x/a\cos\theta_0)$ $\bar{c}/c_0 = 2\cos\theta_0/(1 + \cos^2\theta_0)$
$c_0(1 \mp z/a)$	$(z/a + 1)^2 + (x/a - \tan\theta_0)^2 = \sec^2\theta_0$
$z < 0\ (> 0)$	$\bar{c}/c_0 = \tan\theta_0/\ln(\sec\theta_0 + \tan\theta_0)$

Input to RAY1 consists of a control or switch variable indicating which SVP is to be used, a set of parameters defining the medium (e.g., axial sound speed, vertical scale factor, and gradients with respect to range), a set of initial ray conditions (depth and slope), and parameters for the range step size and maximum range. Output is in the form of a summary table listing the ray coordinates at each vertexing point. These are also called ray turning points. For RR rays, all turning points are characterized by $dz/dx = 0$. (In RAY2 the surface reflection for RSR (Refracted-Surface-Reflected) rays leads to a discontinuous slope.) In addition to the coordinates, RAY1 also lists the maximum depth reached, the cycle distance, the phase integral evaluated over each half-cycle, and the average horizontal ray velocity. Examples are presented in section VI.

Although RAY1 has established that the AIA works well even in severely range-dependent SVPs of certain kinds, one should note that only linear variations of the depth scaling factor have been programmed. An extremely interesting question, which could be answered by minor modifications in RAY1, involves the conservation of J in two range-independent SVPs which are connected by some inhomogeneous region where a(x) varies smoothly. In each range-independent region the phase space trajectories are closed, and J is well defined as the area enclosed by the phase space orbit. As RAY1 is currently written, the trajectories are never closed (unless the user inputs $da/dx = 0$ as a condition). A second question is the effect of nonsymmetric SVPs. With minor modifications, RAY1 could be used to study other more realistic SVPs than those presently programmed.

C. GENERAL DESCRIPTION OF RAY2

RAY2 consists of four Fortran routines which trace a family of rays in a user supplied SVP, computing for each ray the turning point coordinates (depth, range, and time), the

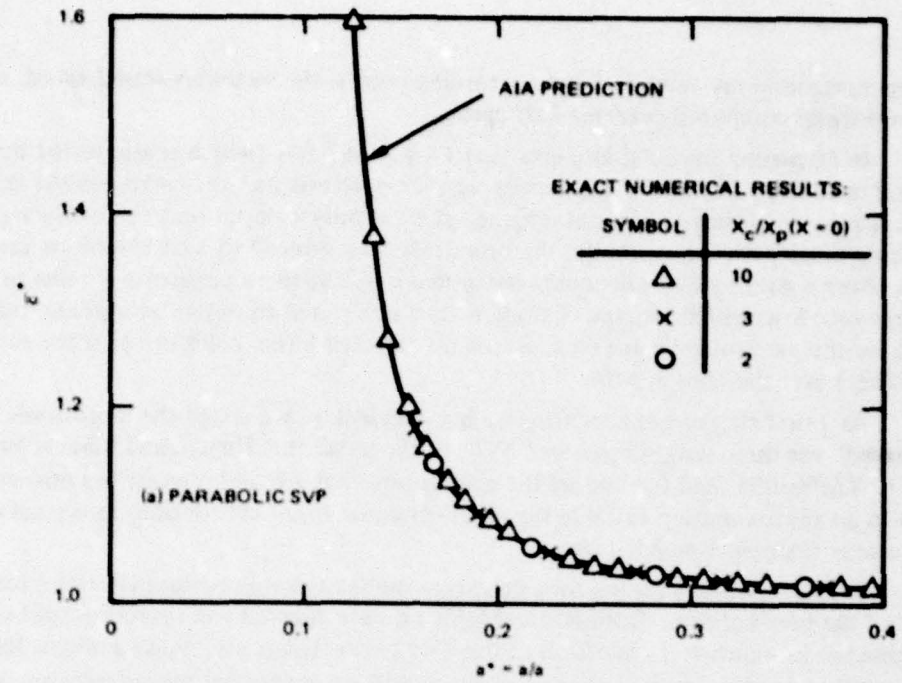


Figure 2. Comparison of AIA predictions and exact numerical results for parabolic SVP (from ref. 2).

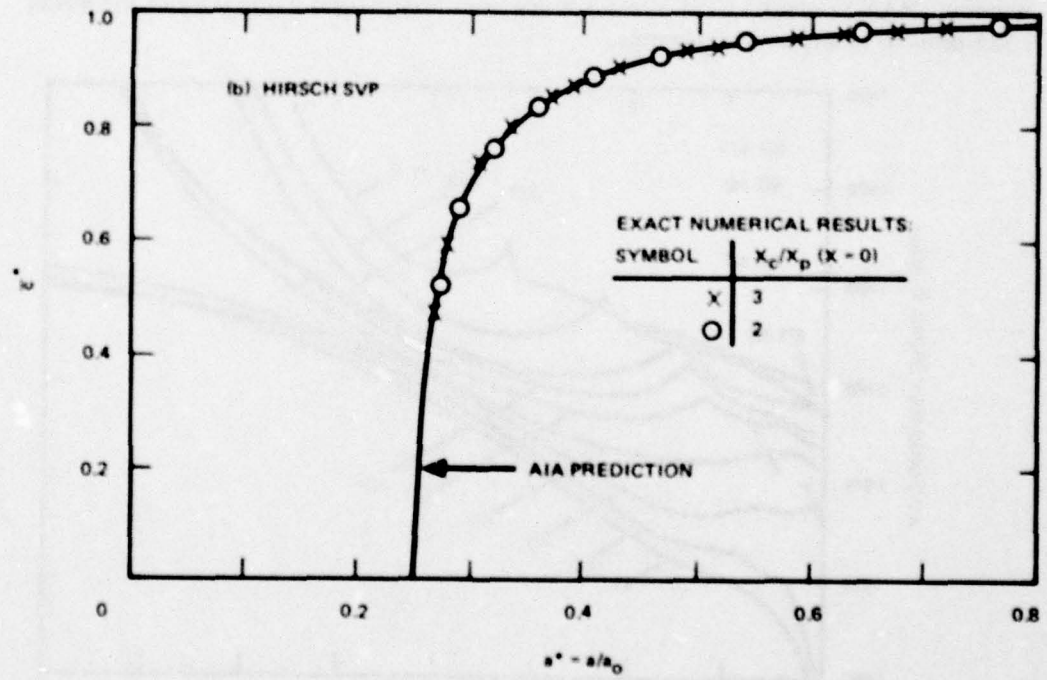


Figure 3. Comparison of AIA predictions and exact numerical results for a Hirsch SVP (from ref. 2).

average horizontal ray velocity between turning points, the vertexing sound speed, and the action integral computed over one-half cycle.

In discussing the adiabatic invariant $J = \oint (\sin\theta)\, dz/c$ (which is also called the action integral in analogy to classical mechanics, or the phase integral since it represents an area in phase space) we outlined a general scheme for efficiently computing an effective signal speed over long-range paths. Essentially, the procedure was reduced to a table-look-up method to obtain from a data base of previously computed $\bar{c}$ vs J tables a particular $\bar{c}$ value in each SVP region. A weighted average of these is then computed to obtain an average sound speed for the particular J value (that is, for the selected initial conditions for the ray trajectory) over the chosen path.

As a first step in implementing such an algorithm, we tested the hypothesis "J is conserved" for three range-dependent SVPs of the parabolic, Hirsch, and bilinear types using RAY1. The results tend to support the assumption that J is an invariant in range-dependent SVPs to an approximation suitable for practical signal speed calculations in typical ocean environments supporting RR rays.[2]

RAY2 provides a means for calculating the relationship between $\bar{c}$ and J for arbitrary SVPs of the user's choice. Both RR and RSR rays are allowed and typical output can be presented as in figure 4. In addition to the $\bar{c}$ vs J curves, one also needs a means for selecting an interval of J values characterizing the rays which carry most of the acoustic signal energy. This may be done by eliminating rays which exceed a cut-off depth or by including only those rays launched from the source (or arriving at the receiver) within a fixed angular window. RAY2 provides a list of ray turning point depths and a summary of the initial conditions to facilitate this selection.

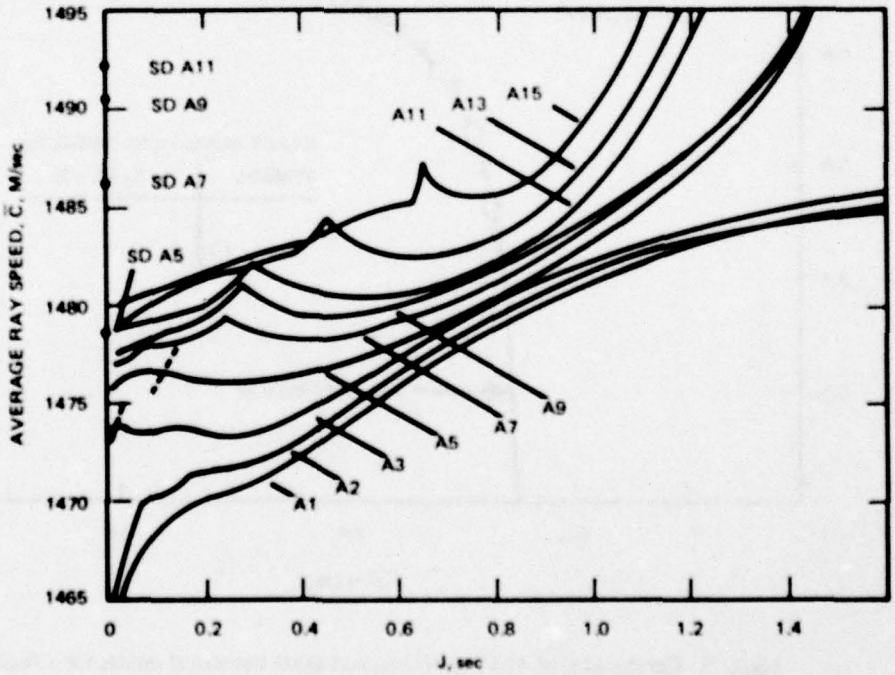


Figure 4. Relation between $\bar{c}$ and J for a selection of SVPs (ref. 2).

Further discussion of RAY2, including input data requirements and internal logic, will be found in sections IV and V. It is worthwhile to stress here that RAY2 treats rays only in homogeneous (range independent) SVPs, and traces a family of 40 rays covering 0.5° to 20° in depression angle at the source out to the fifth vertexing point. (The latter parameters are simply adjusted by internal code changes.) Rays are traced over several half-cycles, even though a single half-cycle suffices to find $\bar{c}$ and J, in order to check the accumulation of round-off errors in the integration routine. Since one is free to refine the mesh size, the trajectory may be made to reproduce itself on each loop to any accuracy (subject to machine constraints). One should use as large a step size as is compatible with the accuracy desired in order to minimize computations.

III. INSTRUCTIONS FOR USING RAY1

Here we describe the required input data, output format, and general interpretation of results for RAY1.

A. INPUT REQUIREMENTS

Input data is read from cards under the Fortran free-field format. This format specification requires commas between separate data items but allows the items to be positioned arbitrarily on a data card. If a single read statement causes the input of data continued on to subsequent cards, no comma is required after the last item on each card. In this case, the card end itself serves as a legal delimiter between successive items.

On the first card, the integer variable ICASE must be punched. This should be the number of rays to be traced.

On the second card, and the remaining ICASE-1 cards, the data items as shown in table 2 must be punched. Typically one card is adequate to enter all items for a ray.

B. OUTPUT

RAY1 allows a quick check of the input data, by printing the variables for each case (each ray) considered in the form

```
CASE 1
DX = 1.2-02 METERS
XMAX = 1.180000+01 M
GRAD 1 = - 6.3510-02 (CHANNEL WIDTH GRADIENT)
GRAD 2 = 0.0000 Hz (AXIAL SPEED GRADIENT IN RANGE)
CO = 1.0000+00 M/SEC
AO = 1.0000+00 M
ZDOT = 3.6397 - 01 RADIANS or 2.0000+01 DEGREES
ZO = 0.0000 M
PROFILE: 1 (1=HIRSCH, 2=PARABOLIC, 3=BILINEAR)
```

If the mesh was refined during any cycle, the next line(s) will present this information in the form

```
MESH = 2 AT STEP 814 .
```

Table 2.

Card	Fortran Variable Name	Definition in RAY1
1	ICASE	Number of rays to be traced
2	DX	Range increment for Runge-Kutta raytracing routine (meters)
	XMAX	Cut-off range for raytracing (meters)
	GRAD1	Gradient in vertical scale length a(x)
	GRAD2	Gradient in axial sound speed $c_0(x)$ (Hz)
	CO	Initial axial sound speed (m/sec)
	AO	Initial vertical scale factor (meters)
	ZDOT	Initial ray slope ($\tan\theta_0$)
	ZO	Initial ray depth (meters)
	IPRINT	Control variable for printing a detailed path (0 : no ; 1 : yes)
	ITYPE	Control variable for SVP type (1 : Hirsch ; 2 : parabolic ; 3 : bilinear)
3	DX,...,ITYPE	Data for next ray

indicating one-half the input step size was used past step 814. These lines will not appear if no step size adjustment was made.

RAY1 employs a cut-off not only in range, but also in the number of steps calculated (presently 9999 steps are done) and in the channel width parameter. The user may inadvertently set up a constricting sound channel such that a(x) becomes zero and then negative in the allowed range. Raytracing is meaningless for such a profile. Rather than "bombing" under such conditions, RAY1 will test a(x), find it is nonpositive, print a message of the form

CHANNEL WIDTH LESS THAN 0.01 AT STEP 704 ,

terminate raytracing of the present ray, and go on to the next case, if there is one.

After RAY1 has completed tracing a given ray to XMAX (or for 9999 steps, or until $a(x) \leq 0$, whichever occurs first), it prints the summary, beginning with the number of axis crossings. *(In contrast to RAY2 where the ocean surface is at $z = 0$, RAY1 always sets $z = 0$* at the channel axis.) Then, for each axis crossing, the following data are printed:

STEP NO	RANGE (M)	TIME (SEC)	AVG(DX/DT) (M/SEC)	A/AO	MAX Z (M)	XP/AO	J (SEC)

Here AO denotes the initial value of the channel width parameter, and A/AO the dimensionless local value of a(x). This is the same function a(x) appearing in eq (12) above. For constricting channels, the numbers in the A/AO column should be monotonically decreasing.

It is worthwhile noting that the AVG(DX/DT) column is derived from pairs of axis crossings and the first value corresponds to the average range rate or horizontal velocity between the initial ray position and the first turning point. For this reason, it may not represent an average between turning points. For the same reason, the last value is arbitrarily set to zero, there being no point beyond the last turning point with which to calculate $\bar{c}$. Similar comments apply to the entries in the J column, the (dimensionless) cycle period column XP/AO, and the maximum depth column.

IV. INSTRUCTIONS FOR USING RAY2

In this section we present information required to understand and run RAY2, beginning with a brief description of the necessary input parameters and the appropriate format for these, the output, and details of the program's scope and internal logical structure.

A. INPUT PARAMETER LIST

RAY2 consists of routines for tracing rays in user-provided SVPs. A cubic spline[6] is used to smooth the input data and to calculate interpolated values and derivatives of c(z) required by the Runge-Kutta routine. Output is provided in a summary form similar to that of RAY1. Detailed instructions for using RAY2 are provided in section IV, and section V contains a listing on the Fortran source code. All input data must be read from cards. Since freefield format is used for input (with one exception, an alphanumeric header card) on may simply place data items, in sequence, in any fields on the cards. Commas must appear between successive items, except when the next card is a continuation of the same read list items. This often occurs for the SVP data. In this case, the end of the card itself acts as a legal delimiter between the last item on the given card and the first item on the next card, and no comma is required.

RAY2 requires the input shown in table 3 immediately after the usual @XQT RAY2.ABS statement.

The variable NOSVPS tells RAY2 how many SVP functions are to follow. If this card is incorrectly punched (or misread), RAY2 will either end its computations before reaching the last data set or attempt to read beyond the last data set provided. If this card is missing, RAY2 will recognize NPTS as NOSVPS, and then fail (probably with a fatal execution error) when attempting to read card 3 under the format 2I10,2A6.

On card 2, NPTS is the number of pairs of depth (ZSVP) and sound speed (CSVP) values comprising the first SVP. If NPTS is mispunched RAY2 may attempt to read past the last card, or may stop before reading the last card, and proceed to recognize card j+1 as SVP data or SVP data as card j+1. Results of such input errors could include many forms of abnormal termination.

Immediately following NPTS is IUNITS, which indicates the choice of units made by the user for SVP data. Set IUNITS equal to 1 if the units are to be meters and meters per second and 2 if feet and feet per second.

---

6. Mathpac, Univac Corp. (SPLN1 and SPLN2 are documented in a number of manuals covering resident subroutines.)

Table 3.

Card Number	Fortran Variable or Array Name	Input Format Spec (Variable Type)
1	NOSVPS	Free Field (Integer)
2	NPTS, IUNITS, MSG(1) MSG(2)	2I10,2A6
3	ZSVP(1), CSVP(1) ... ZSVP(i), CSVP(i)	Free Field (Real)
4	ZSVP(i+1), CSVP(i+1),	Free Field (Real)
. . .	...	
j	... ZSVP(NPTS), CSVP(NPTS)	Free Field (Real)
j+1	DX, ZO, ZMAX, IPLOT	Free Field (Real, Real, Real, Integer)
j+2, . . . k	data for next SVP, beginning with NPTS, IUNITS, MSG(1), MSG(2), and ending with DX, ZO, etc.	
k+1, etc.	data for last of NOSVPS data sets	

MSG(1) and MSG(2) provide a title for the SVP to follow, consisting of 12 characters selected from the full range of UNIVAC 1110 Fortran and entered in columns 21-32 of card 2.

Note that card 2 is read under a formatted input specification.

Beginning with card 3, and continuing on subsequent cards if necessary, is the first set of SVP data. As many pairs may be entered on one card as space permits and pairs of (ZSVP, CSVP) values may be split, with ZSVP(i) ending one card and CSVP(i) beginning the next card.

Four variables which control raytracing are entered on the card after the last SVP data card. DX, ZO, and ZMAX are the range increment for the Runge-Kutta algorithm, the initial ray depth, and the maximum depth allowed for a ray, respectively. *These variables are interpreted by RAY2 as being expressed in meters.* IPLOT should be set to 0 if one desires printed output, and 1 if plots are desired in addition. Typical values for DX should be approximately one-sixteenth of a ray cycle period, for coarse results, and less for more accurate raytracing. If a ray exceeds depth ZMAX, the remaining (steeper) rays are not traced and RAY2 attempts to read the next SVP data set.

B. OUTPUT

RAY2 is designed to allow the user an immediate check of all input. The first line printed after the @XQT statement should agree with the input variable NOSVPS, and reads:

*** THIS RUN CONSISTS OF (NOSVPS) SVP(S) ***

The second output line consists of the 12 character message string MSG(1), MSG(2) imbedded as follows:

*** SVP : (MSG(1), MSG(2)) ***

This simply identifies the particular SVP being computed. If several SVPs are to be run, one such header message appears at the start of the output for each SVP.

The header message is followed immediately by a line that checks the choice of units for the SVP data, and reads:

USER INPUT UNITS FOR THE SVP : (1 or 2) (1 : M, M/SEC ; 2 : FT, FT/SEC)

IUNITS and the units choice (1 or 2) should agree. Next the SVP is printed as a pair of columns labeled DEPTH and c. There should be NPTS pairs of values.

After the SVP data, RAY2 prints the control parameters specifying how rays are to be traced, namely DX, ZO, and ZMAX.

Beginning with the line

CASE NUMBER : 1 ,

RAY2 prints, as presently structured, 40 ray-path summaries, for the rays which are launched from depth ZO at the angles -0.5°, -1.0°, -1.5°, ..., -20°, with respect to horizontal. Each case is summarized by a set of initial conditions printed in the form

```
CASE NUMBER : 1
INITIAL ANGLE = 0.5 DEGREES
INITIAL SOUND SPEED = 1495.6 M/SEC
PREDICTED VERTEXING SPEED = 1500.2 M/SEC
```

The first and second lines are redundant since the initial angle associated with a given case is always one-half the case number. Nevertheless, it is useful to see at a glance the angle of the ray being considered, and if internal changes are made to produce a different angular interval for rays, these two lines would serve as checks on the code changes and would simplify some data analysis.

The initial sound speed is computed by the spline curve fitting routine SPLN1,[6] and used to compute a predicted vertex speed according to Snell's law,

$$c_{vertex} = c(z_0)/\cos\theta\ (z_0) \ .$$

Each case is also summarized by a table showing the history of ray turning points. As presently written RAY2 automatically traces rays through four turning points. The variables printed at each turning point (with the exception of J for the first turning point and the average speed for the last turning point) are:

K, XTP, TTP, ZTP, CTP, J(K), AVGSP .

These stand for the turning point number, K, the range to the Kth turning point, the time elapsed at the Kth turning point, the depth of the Kth turning point, the ray speed at the Kth turning point, the action integral $\int \sin\theta \, dz/c$ computed over the last one-half cycle, and the average horizontal ray speed from the Kth to the (K+1)st turning point.

Figure 5 illustrates the physical significance of these summary data items. Note that the first value printed in the J column is really an incomplete integral, evaluated from the starting depth z_0 to the first turning point, and should be ignored. RAY2 computes the average horizontal ray velocity between pairs of turning points. The first AVGSP value printed out is associated with the first and second turning points, and the fourth value is automatically set to zero.

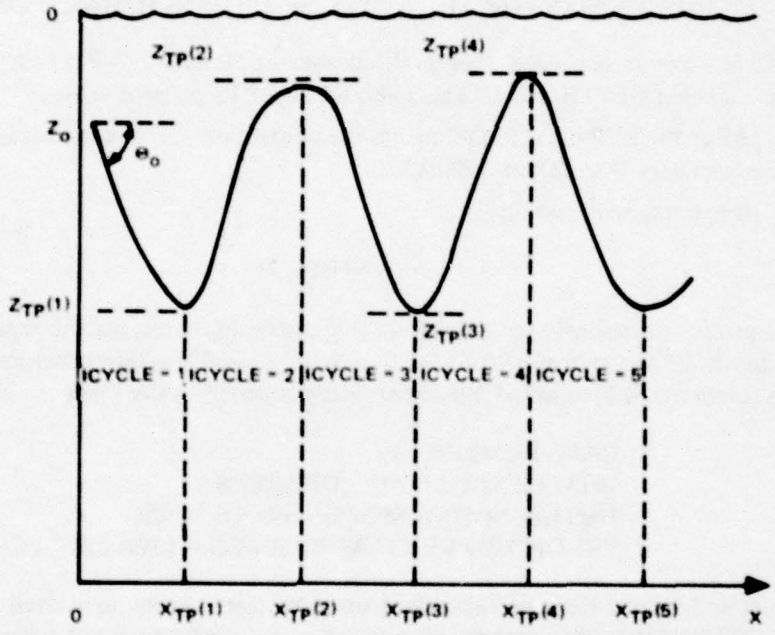


Figure 5. A sketch of the physical meaning of the summary data items in RAY2.

V. INTERNAL LOGICAL STRUCTURE OF RAY1 AND RAY2

In this section we shall describe the logical structure of the Fortran routines comprising RAY1 and RAY2. Special emphasis is placed on the areas in which slight modifications in these codes would enhance their usefulness, particularly in the direction of permitting better output (e.g., plots) and more flexibility through user-selected input variables. beyond those now available.

A. INTERNAL STRUCTURE OF RAY1

As it is presently written, RAY1 contains comment cards defining most of the variables of interest (see section VI). A flowchart of RAY1 appears in figure 6. ICASE is

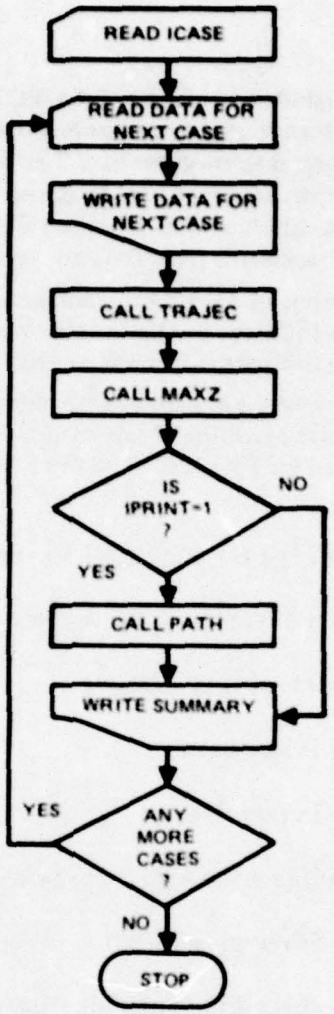


Figure 6. Flowchart of RAY1.

the number of cases to be considered. Each case consists of a particular choice for an SVP, initial ray depth and slope, and parameters for the range step, maximum range, and gradients of the range-dependent parameters (axial sound speed and channel width parameter in eq (12)).. In addition, RAY1 reads IPRINT which allows or suppresses a longer printout of the trajectory.

An extremely valuable addition to RAY1 would be a plotting routine which simply plots the ray paths. One could then observe immediately (without calling the subroutine PATH) the effect of squeezing or expanding SVPs on ray paths. A second much needed improvement consists of permitting several rays to be traced in a given range-dependent SVP, without the requirement of punching all SVP data and other parameters anew for each ray.

The subroutine TRAJEC computes a ray path, including the time and average horizontal velocity at each turning point, and the action integral over each half-cycle. Beginning at line 26 (see section VI) TRAJEC computes the new ray slope (DERIZ). These instructions end at line 40 of TRAJEC, and consist of the Runge-Kutta algorithm mentioned earlier.

Once an advanced value of z is found, TRAJEC finds the new sound speed at this depth and range, which in turn is used to compute a time increment given by $\Delta t = \Delta s/c$, and a new increment of the action integral given by $c^{-1} \sin\theta \, \Delta z$. These steps correspond to lines 42 through 55. Two tests are made here to insure that the user has not allowed the channel width parameter to vanish, and to avoid any illegal (or rather invalid) reference to array elements by looking backwards from the very first step of the raytracing.

The remaining portions of TRAJEC handle axis crossings and adjust the step size should there be fewer than 150 steps per half-cycle. We recommend that an input variable be added to permit varying this particular choice at run time.

In section II we derived the Euler-Lagrange equation, eq (7), for Fermat's variational principle. RAY1 calculates the right-hand side of this equation of motion in the function subroutine F, which is called by TRAJEC. The three SVPs in equations (12) lead, respectively, to the results

$$F = -2(1 + \dot{z}^2)\, z/(a^2 + z^2) + \dot{z}(1 + \dot{z}^2)\, g_2/c_0 - 2\,\dot{z}(1 + \dot{z}^2)\,(z^2/a)\, g_1/(a^2 + z^2) \,, \tag{13a}$$

$$F = -(z/a)^2 (1 + \dot{z}^2)/(1 - (z/a)^2) + \dot{z}(1 + \dot{z}^2)\, g_2/c_0 - \dot{z}(1 + \dot{z}^2\,(z^2/a^3)\, g_1/(1 - z^2/a^2) \,, \tag{13b}$$

$$F = \pm(1 + \dot{z}^2)/(a + |z|) + \dot{z}(1 + \dot{z}^2)\, g_2/c_0 - \dot{z}(1 + \dot{z}^2)\, |z|\, g_1/a(a + |z|) \quad , \quad z \lessgtr 0 \;(>0) \;, \tag{13c}$$

as one can verify easily by direct differentiation. Here g_1 and g_2 are respectively da/dx and $dc(z=0)/dx$.

MAXZ and PATH are short routines. The former searches each half cycle of the trajectory and returns a list of the maximum off-axis distance reached by the ray. The latter prints 250 points equally spaced along the trajectory, or as many points as are found before the maximum range is reached.

RAY1 should be expanded to handle a wider class of range-dependent analytic SVPs, and to permit connecting range-independent SVPs smoothly, as discussed in section II.

B. INTERNAL STRUCTURE OF RAY2

RAY2 consists of only two routines and two cubic spline routines resident on the NOSC UNIVAC 1110.[6] Unlike RAY1, RAY2 does provide a plot subroutine and uses any user-provided SVP. We recommend that the plot routine calls be modified to allow several rays to be plotted on a single graph. Also desirable would be a plot of the user-input SVP as smoothed by the spline routines.

A flowchart of RAY2 is given in figure 7. Although somewhat complicated in appearance, the flowchart involves only three basic loops. The outermost, which corresponds

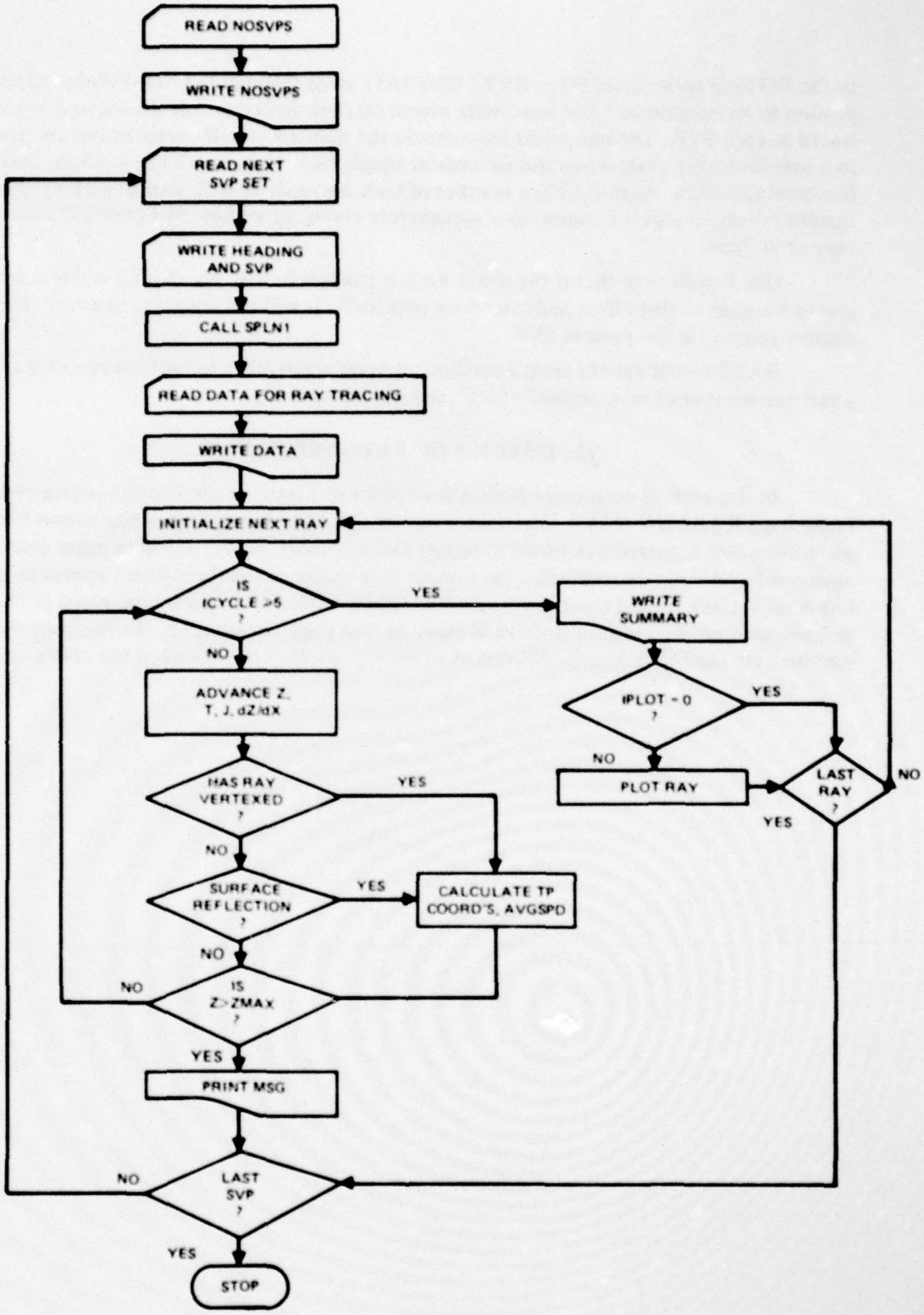


Figure 7. Flowchart of RAY2.

to the DO loop with object 90 in RAY2 (line 163), steps through the NOSVPS sound speed profiles to be considered. The next, with object 60 (line 162) controls the 40 rays to be traced in each SVP. The innermost loop checks the number of half-cycles traced and jumps to a new initial ray angle when this exceeds or equals five. Hence each ray is traced over five turning points. As in RAY1, a number of tests are made at each step to pick up the turning points, except, of course, that surface reflections, as well as total internal reflections, may occur here.

One should note that if the depth limit is reached by any ray, RAY2 will automatically print a message to this effect and look for a new SVP. It will not trace the next ray (0.5 degrees steeper) in the present SVP.

RAY2 would benefit from a revision allowing a variable spacing between rays and a variable number of rays, instead of 0.5° and 40, respectively.

VI. LISTINGS OF RAY1 AND RAY2

In this section we present listings (computer printout) of the Fortran source code comprising RAY1 and RAY2. The reader should note that in some cases pagination becomes awkward when converting printout to report format. In particular, separate pages cannot always refer to separate routines. The Fortran line sequence numbers which appear in the left-hand column should therefore be used to follow a routine across several pages or to distinguish multiple routines from each other on one page. In addition, the line sequence numbers are used elsewhere in this report to refer to particular sections of the codes.

```
00100      1*      C  THIS PROGRAM FINDS RAY TRAJECTORIES IN A HIRSCH, PARABOLIC, OR                  000000
00100      2*      C  BILINEAR PROFILE WHOSE AXIAL SOUND SPEED AND CHANNEL WIDTH ARE                  000000
00100      3*      C  FUNCTIONS OF RANGE.                                                             000000
00100      4*      C  THE VARIABLES ARE:                                                              000000
00100      5*      C     CO,AD=INITIAL AXIAL SOUND SPEED AND CHANNEL WIDTH (M/SEC AND M)             000000
00100      6*      C     ZO,ZOOT=INITIAL RAY DEPTH AND SLOPE (M)                                      000000
00100      7*      C     GRAD1,GRAD2=GRADIENTS IN A AND C(Z=0) (DIMENSIONLESS AND IN HZ)              000000
00100      8*      C     Z=ARRAY OF DEPTHS, DR=RANGE INCREMENT, N=NUMBER OF RANGE STEPS               000000
00100      9*      C     T=ARRAY OF ELAPSED TIMES                                                     000000
00100     10*      C     J=ARRAY OF POINTERS TO RAY AXIS CROSSING RANGES                              000000
00100     11*      C     AV=ARRAY OF AVERAGE HORIZONTAL SPEEDS FOR EACH CYCLE                         000000
00100     12*      C     ICOUNT=NUMBER OF AXIS-CROSSINGS (INCLUDING THE STARTING POINT)               000000
00100     13*      C     ICASE=NUMBER OF CASES                                                        000000
00100     14*      C     IPRINT=CONTROL VARIABLE FOR PRINTING TRAJECTORY (1:PRINT, 0: NO)             000000
00100     15*      C     ITYPE=CONTROL VARIABLE FOR RAY TYPE (1=HIRSCH,2=PARABOLIC,                   000000
00100     16*      C           3=BILINEAR)                                                            000000
00100     17*      C     R0,T0,A=ARRAYS CONTAINING THE AXIS-CROSSING RANGES,TIMES AND A VALUES        000000
00100     18*      C     N=NUMBER OF STEPS NEEDED TO REACH MAX RANGE, OR A 100-FOLD REDUCTION         000000
00100     19*      C         IN CHANNEL WIDTH                                                         000000
00100     20*      C     ACTION=ARRAY OF INTEGRALS OVER EACH HALF-CYCLE OF THE ACTION                 000000
00100     21*      C                                                                                  000000
00101     22*            COMMON/DATA/DR,XMAX,GRAD1,GRAD2,CO,AD,ZOOT,ZO,ITYPE                          000000
00103     23*            COMMON/RESULT/Z(130001,T(100001,R(100001,DER(Z(100001,ICOUNT,                000001
00103     24*           *   J(7001,AV(2001,XMAX(2001,ACTION(2001,R0(2001,T0(2001,A(2001,N             000001
00103     25*      C                                                                                  000001
00104     26*            READ(5,10)ICASE                                                              000001
00107     27*            DO 30 IC=1,ICASE                                                             000007
00107     28*      C                                                                                  000007
00107     29*      C  READ AND WRITE DATA                                                             000007
```

```
      READ(5,101)DT,XMAX,GRAD1,GRAD2,C0,A0,ZDOT,Z0,IPRINT,ITYPE
  101 FORMAT( )
      THETA=ATAN(ZDOT)*180./3.14159265
      WRITE(6,102)IC,DT,XMAX,GRAD1,GRAD2,C0,A0,ZDOT,THETA,Z0,ITYPE
  102 FORMAT(1H1,'CASE ',I3/
     *   ' DX=',1PE10.4,' METERS'/' XMAX=',1PE12.6,' M'/' GRAD1=',
     *1PE10.4,'(CHANNEL WIDTH GRADIENT)'/' GRAD2=',1PE10.4,' HZ (AXIAL
     *SPEED GRADIENT IN RANGE)'/' C0=',1PE10.4,' M/SEC'/
     *' A0=',1PE10.4,' M'/' ZDOT=',1PE10.4,' RADIANS OR',1PE10.4,
     *' DEGREES'/' Z0=',1PE10.4,' M'/' PROFILE:',I3,' (1=HIRSCH,2=PARABO
     *LIC,3=BILINEAR)'///' RESULTS:'/)
C
C  FIND RAY PATH, TIMES, AVERAGE SPEEDS AND MAXIMUM RAY EXCURSION.
C
      CALL TRAJEC
      ICNTM1=ICOUNT-1
      WRITE(6,105)ICNTM1
  105 FORMAT(1H0,' AXIS CROSSINGS:',I5)
      CALL MAXZ
      IF (IPRINT .EQ.1)CALL PATH
C
C  WRITE SUMMARY OF AXIS-CROSSINGS
C
      WRITE(6,103)
  103 FORMAT(1H0,' SUMMARY OF AXIS CROSSINGS:'/' NUMBER  STEP',
     *    9X,'RANGE',10X,'TIME',6X,'AVG(DR/DT)',5X,'A/A0',7X,'MAX Z',
     *    10X,'XP/A0',8X,'J'/10X,'NO.',5X,'(M)',11X,'(SEC)',6X,'(M/SEC)',
     *    20X,'(M)',21X,'(SEC)')
      DO 10 I=1,ICNTM1
         XP=X0(I+1)-X0(I)
   10    WRITE(6,104)I,J(I),X0(I),T0(I),AV(I),A(I),ZMAX(I),XP,ACTION(I)
  104    FORMAT(2I6,2X,7(1PE12.4,1X))
      TEMP=0
      I=ICOUNT
      WRITE(6,104)I,J(I),X0(I),T0(I),TEMP,A(I),TEMP,TEMP,TEMP
C
   30 CONTINUE
C
      STOP
      END
```

```
00101         SUBROUTINE TRAJEC                                                   000000
00101   C  THIS SUBROUTINE USES A FOURTH ORDER ACCURATE RUNGE-KUTTA SCHEME TO      000000
00101   C  INTEGRATE THE EQUATION OF MOTION FOR A RAY:                             000000
00101   C     D(DZ/DX)/DX=-(1+(DZ/DX)**2)*(DC/DZ)/C                                000000
00101   C                 +DZ/DX*(1+(DZ/DX)**2)*(1/C)*DC/DX                        000000
00101   C  DC/DZ AND DC/DX ARE PARTIAL DERIVATIVES, Z IS DEPTH, X RANGE, AND       000000
00101   C  C THE LOCAL SOUND SPEED.  THIS SUBROUTINE ALSO COMPUTES THE TIME AND A  000000
00101   C  AVERAGE SPEED ALONG THE PATH, AND THE INTEGRAL OF SIN(THETA)/C          000000
00101   C  WITH RESPECT TO Z OVER EACH HALF-CYCLE.                                 000000
00101   C  THIS SUBROUTINE ALSO CUTS THE STEP SIZE IN HALF WHEN THERE ARE 50       000000
00101   C  OR FEWER STEPS PER HALF-CYCLE.                                          000000
00103         COMMON/DATA/FX,XMAX,GRAD1,GRAD2,C0,A0,ZDOT,Z0,ITYPE                 000000
00104         COMMON/RESULT/Z(10000),T(10000),X(10000),DERIZ(10000),ICOUNT,       000000
00104      *    J(200),AV(200),ZMAX(200),ACTION(200),R0(200),T0(200),A(200),N     000000
00105         REAL K1,K2,K3,K4                                                    000000
00106         Z(1)=Z0                                                             000000
00107         X(1)=0.                                                             000001
00110         DERIZ(1)=ZDOT                                                       000002
00111         T(1)=0.                                                             000004
00112         R0(1)=0.                                                            000005
00113         T0(1)=0.                                                            000006
00114         A(1)=A0                                                             000007
00115         ICOUNT=1                                                            000011
00116         J(1)=1                                                              000013
00117         MESH=1                                                              000014
00120         DO 10 I=1,9999                                                      000020
00123            IF (I.EQ.ICYCLE)=0                                               000020
00125            IF (I.GT.1)X(I)=X(I-1)+DX                                        000024
00127            IF (X(I).GE.XMAX)GO TO 20                                        000033
```

```
00131   300            Y=Z(I)                                                  000037
00132   310            YP=DERIZ(I)                                             000041
00133   320            W=R(I)                                                  000043
00134   330            K1=DS*F(W,Y,YP)                                         000045
00135   340            K2=DS*F(W+DS/2,Y+DS*YP/2+DS*K1/8,YP+K1/2)               000054
00136   350            K3=DS*F(W+DS/2,Y+DS*YP/2+DS*K2/8,YP+K2/2)               000103
00137   360            K4=DS*F(W+DS,Y+DS*YP+DS*K3)/2,YP+K3)                    000132
00137   370    C                                                               000132
00140   380            IP1=I+1                                                 000157
00141   390            Z(IP1)=Z(I)+DS*(YP+(K1+K2+K3)/6.)                       000163
00142   400            DERIZ(IP1)=YP+(K1+2*K2+2*K3+K4)/6.                      000175
00142   410    C                                                               000175
00143   420            IF (I.EQ.1)GO TO 10                                     000207
00143   430    C  IF THIS IS THE FIRST STEP, DO NOT COMPUTE TIME OR AVERAGE SPEED  000207
00143   440    C                                                               000207
00146   450            ATEMP=A0+GRAD1*Z(I)                                     000212
00146   460            IF (ATEMP.LE.0.0)GO TO 30                               000216
00150   470            C=CO+GRAD2*Z(I)                                         000220
00151   480            DS=SQRT(DS*DS+(Z(I-1)-Z(I))**2)                         000226
00152   490            IF (ITYPE.EQ.1)C=C/SQRT(1.-(Z(I)/ATEMP)**2)             000241
00154   500            IF (ITYPE.EQ.2)C=C*(1+(Z(I)/ATEMP)**2)                  000256
00156   510            IF (ITYPE.EQ.3)C=C*(1+ABS(Z(I)/ATEMP))                  000266
00156   520    C                                                               000266
00160   530            T(I)=T(I-1)+DS/C                                        000277
00161   540            ACTION(ICOUNT)=ACTION(ICOUNT)+(1/C)*(Z(I)-Z(I-1))*      000303
00161   550     *          SIN(0.5*(ATAN(DERIZ(I))+ATAN(DERIZ(I-1))))          000303
00161   560    C  IF AN AXIS CROSSING HAS OCCURRED, LABEL THE POINT AND CALCULATE THE  000303
00161   570    C  AVERAGE SPEED, AND ADJUST STEP SIZE IF NECESSARY             000303
00162   580            IF (Z(I).EQ.0) GO TO 10                                 000335
00164   590            IF (Z(I-1)*Z(I).GT.0) GO TO 10                          000337
00166   600            IF (I.EQ.2) GO TO 10                                    000343
00166   610    C                                                               000343
00170   620            ICOUNT=ICOUNT+1                                         000346
00171   630            J(ICOUNT)=I                                             000352
00172   640            G=Z(I-1)/(Z(I-1)-Z(I))                                  000354
00173   650            X0(ICOUNT)=X(I-1)+G*(X(I)-X(I-1))                       000361
00174   660            T0(ICOUNT)=T(I-1)+G*(T(I)-T(I-1))                       000366
00175   670            A(ICOUNT)=A0+GRAD1*(X(I-1)+G*DX)                        000373
00176   680            ICNTM1=ICOUNT-1                                         000401
00177   690            AV(ICNTM1)=(X0(ICOUNT)-X0(ICNTM1))/(T0(ICOUNT)-T0(ICNTM1))  000404
00177   700    C                                                               000404
00177   710    C  ADJUST STEP SIZE IF THERE ARE LESS THAN 150 STEPS PER HALF-CYCLE  000404
00177   720    C                                                               000404
00200   730            IF (J(ICOUNT)-J(ICNTM1).GT.150)GO TO 10                 000410
00200   740    C                                                               000410
00202   750            DS=DS/2                                                 000414
00203   760            MESH=MESH+1                                             000417
00204   770            WRITE(6,100)MESH,I                                      000422
00210   780      100   FORMAT(1HO,' MESH=',I3,' AT STEP',I5)                   000433
00211   790       10 CONTINUE                                                  000433
00213   800          N=I                                                       000433
00214   810          RETURN                                                    000435
00215   820       30 N=I                                                       000441
00216   830          WRITE(6,101)N                                             000442
00221   840      101 FORMAT(1HO,' MAXIMUM DISTANCE AT STEP',I5/)               000451
00222   850       30 N=I                                                       000451
00223   860          WRITE(6,102)N                                             000452

00227A  870      102 FORMAT(1HO,' CHANNEL WIDTH LESS THAN 0.01 AT STEP',I5)    000460
00227   880          RETURN                                                    000460
00230   890          END                                                       000511
```

```
@FOR,S RAY.PATH
FOR 5R3-02/15/78-11:08:41 (7,)

  SUBROUTINE PATH      ENTRY POINT 000046

  STORAGE USED: CODE(1) 000072; DATA(0) 000047; BLANK COMMON(2) 000000

   COMMON BLOCKS:

   0003   DATA    000011
   0004   RESULT  120622

  EXTERNAL REFERENCES (BLOCK, NAME)

   0005   NWDU$
   0006   N102$
   0007   NERR3$

                                                                                000002
00101      1*          SUBROUTINE PATH                                          000002
00103      2*          COMMON/DATA/DZ,XMAX,GRAD1,GRAD2,CO,AO,Z00T,ZO,ITYPE      000002
00104      3*          COMMON/RESULT/Z(10000),T(10000),X(10000),DERIZ(10000),ICOUNT,   000002
00104      4*        *    J(200),AV(200),ZMAX(200),ACTION(200),XO(200),TO(200),A(200),N   000002
00105      5*          WRITE(6,101)                                             000007
00107      6*      101 FORMAT(1H0,' RAY TRAJECTORY:'//'  STEP NO.  RANGE',      000007
00107      7*         *5X,'DEPTH',5X,'CHANNEL'4X,'TIME'/                        000007
00107      8*         *12X,'(M)',7X,'(M)',7X,'WIDTH(M)',6X,'(SEC)'/)            000007
00110      9*          K=N/250                                                  000016
00111     10*          DO 10 I=1,N                                              000016
00114     11*          ATEMP=AO+GRAD1*X(I)                                      000022
00115     12*          R=X(I)                                                   000024
00116     13*          IF (R.GE.XMAX) RETURN                                    000033
00120     14*       10 IF(MOD(I,K) .EQ.0) WRITE(6,102)I,R,Z(I),ATEMP,T(I)       000054
00131     15*      102 FORMAT(I6,4X,4(1PE10.4))                                 000054
00132     16*          RETURN                                                   000071
00133     17*          END
```

```
00101        FUNCTION F(X,Y,YP)                                                  000000
00101   C  THIS FUNCTION SUBROUTINE SIMPLY COMPUTES THE RIGHT HAND SIDE OF THE   000000
00101   C  EQUATION OF MOTION                                                    000000
00103        COMMON/DATA/DX,XMAX,GRAD1,GRAD2,CO,AO,ZDOT,ZO,ITYPE                 000000
00104        A=AO+GRAD1*X                                                        000000
00105        IF (ITYPE.EQ.2)GO TO 2                                              000003
00107        IF (ITYPE.EQ.3) GO TO 3                                             000006
00111        F=-(Y/A**2)*(1+YP**2)/(1-(Y/A)**2)                                  000023
00111      *  +YP*(1+YP**2)*GRAD2/CO                                             000023
00111      *  +YP*(1+YP**2)*(Y**2/A**3)*GRAD1/(1-Y**2/A**2)                      000023
00112        RETURN                                                              000050
00113      2 F=-2*(1+YP**2)*Y/(A**2+Y**2)                                        000054
00113      *  +YP*(1+YP**2)*GRAD2/CO                                             000054
00113      *  -2*YP*(1+YP**2)*(Y**2/A)*GRAD1/(A**2+Y**2)                         000054
00114        RETURN                                                              000110
00115      3 IF (Y.LE.0)S=1                                                      000114
00117        IF (Y.GT.0)S=-1                                                     000126
00121        F=S*(1+YP**2)/(A+ABS(Y))                                            000133
00121      *  +YP*(1+YP**2)*GRAD2/CO                                             000133
00121      *  -YP*(1+YP**2)*ABS(Y)*GRAD1/A*(A+ABS(Y))                            000133
00122        RETURN                                                              000153
00123        END                                                                 000175
```

```
00101          SUBROUTINE MAXZ                                                         000000
00101    C  THIS ROUTINE USES THE POINTERS IN ARRAY J TO FIND THE MAXIMUM              000000
00101    C  MAX EXCURSION OVER EACH CYCLE                                              000000
00103          COMMON/RESULT/Z(10000),T(10000),X(10000),DERIZ(10000),ICOUNT,          000000
00103         *   J(200),AV(200),ZMAX(200),ACTION(200),RO(200),TO(200),A(200),N       000000
00104          ICNTM1=ICOUNT-1                                                         000000
00105          DO 10 I=1,ICNTM1                                                        000002
00110          K1=J(I)                                                                 000006
00111          K2=J(I+1)                                                               000010
00112          TEMP=ABS(Z(K1))                                                         000013
00113             DO 20 K=K1,K2                                                        000020
00114     20      IF (ABS(Z(K)).GE.ABS(TEMP))TEMP=Z(K)                                 000020
00121     10   ZMAX(I)=TEMP                                                            000030
00123          RETURN                                                                  000033
00124          END                                                                     000054
```

```
CASE    1

DX= 1.0000-02 METERS
XMAX= 1.1808000+01 M
GRAD1=-6.3517-02(CHANNEL WIDTH GRADIENT)
GRAD2= 0.0000     HZ (AXIAL  SPEED GRADIENT IN RANGE)
CO= 1.0000+00 M/SEC
AO= 1.0000+00 M
ZDOT= 3.6397-01 RADIANS OR 2.0000+01 DEGREES
ZO= 0.0000    M
PROFILE:  1 (1=HIRSCH,2=PARABOLIC,3=BILINEAR)


RESULTS:

 MESH= 2 AT STEP  819

 MESH= 3 AT STEP 1538

 MAXIMUM DISTANCE AT STEP 1562


 CHANNEL WIDTH LESS THAN 0.01 AT STEP 1562

 AXIS CROSSINGS:    8

 SUMMARY OF AXIS CROSSINGS:
```

NUMBER	STEP NO.	RANGE (M)	TIME (SEC)	AVG(DR/DT) (M/SEC)	A/AO	MAX Z (M)	XP/AO	J (SEC)
1	1	0.000000	0.000000	9.974316-01	1.000000+00	3.264505-01	2.676209+00	1.845267-01
2	249	2.676209+00	2.682563+00	9.964728-01	9.300390-01	-2.977873-01	2.190655+00	1.839466-01
3	488	4.866864+00	4.880971+00	9.947468-01	6.909055-01	2.714899-01	1.792687+00	1.834989-01
4	667	6.659551+00	6.683126+00	9.921799-01	5.770519-01	-2.987117-01	1.966395+00	1.849397-01
5	814	8.125976+00	8.161077+00	9.874639-01	4.839212-01	2.781555-01	1.194519+00	1.844375-01
6	1053	9.320464+00	9.370761+00	9.827181-01	4.080573-01	-2.097171-01	9.796389-01	1.846039-01
7	1299	1.030010+01	1.036761+01	9.793444-01	3.958405-01	1.933680-01	7.989769-01	1.824479-01
8	1408	1.109903+01	1.118759+01	9.619762-01	2.951006-01	-1.789361-01	6.501577-01	1.839895-01
9	1538	1.174919+01	1.186343+01	0.000000	2.538091-01	0.000000	0.000000	0.000000

CASE 2
DR= 1.0000-02 METERS
RMAX= 2.3417D0+01 M
GRAD1=-3.1757-02(CHANNEL WIDTH GRADIENT)
GRAD2= 0.0000 M/ (AXIAL SPEED GRADIENT IN RANGE)
C0= 1.0000.00 M/SEC
A0= 1.0000.00 M
ZOOT= 3.6397-01 RADIANS OR 2.0000+01 DEGREES
Z0= 0.0000 M
PROFILE: 1 (1=HIRSCH,2=PARABOLIC,3=BILINEAR)

RESULTS:

MESH= 2 AT STEP 1427

MESH= 3 AT STEP 2952

MAXIMUM DISTANCE AT STEP 3295

CHANNEL WIDTH LESS THAN 0.01 AT STEP 3295

AXIS CROSSINGS: 16

SUMMARY OF AXIS CROSSINGS:

NUMBER	STEP NO.	RANGE (M)	TIME (SEC)	AVG(DR/DT) (M/SEC)	A/A0	MAX Z (M)	ZP/A0	J (SEC)
1	1	0.000000	0.000000	9.978637-01	1.000000+00	3.391714-01	2.809510+00	3.683913-01
2	282	2.809510+00	2.815525+00	9.973737-01	9.107789-01	-3.189649-01	2.592109+00	3.689492-01
3	537	5.351619+00	5.364276+00	9.948199-01	8.300487-01	3.095518-01	2.300038+00	3.673702-01
4	762	7.651657+00	7.671653+00	9.961209-01	7.570043-01	-2.908967-01	2.080886+00	3.687151-01
5	975	9.732592+00	9.760693+00	9.952694-01	6.909237-01	2.779658-01	1.882968+00	3.678670-01
6	1163	1.161501+01	1.165207+01	9.992219-01	6.311921-01	-2.657264-01	1.702808+00	3.677626-01
7	1333	1.331782+01	1.336977+01	9.979984-01	5.770660-01	2.591996-01	1.540115+00	3.662567-01
8	1487	1.485793+01	1.491583+01	9.913782-01	5.291566-01	-2.431976-01	1.392769+00	3.698338-01
9	1627	1.625070+01	1.632068+01	9.884336-01	4.839269-01	2.329394-01	1.259686+00	2.190299-01
10	1757	1.750539+01	1.758979+01	9.871967-01	4.440819-01	-2.231793-01	1.138731+00	1.027189-01
11	2104	1.884912+01	1.893329+01	9.841914-01	4.079187-01	2.139698-01	1.029122+00	1.041937-01
12	2310	1.967324+01	1.978873+01	9.809763-01	3.752369-01	-2.052928-01	9.297681-01	1.839958-01
13	2494	2.060301+01	2.073653+01	9.768227-01	3.457102-01	1.971329-01	8.396776-01	1.890626-01
14	2644	2.144269+01	2.159613+01	9.717774-01	3.190496-01	-1.896620-01	7.579539-01	1.899812-01
15	2816	2.220064+01	2.237610+01	9.654502-01	2.949792-01	1.822616-01	6.837816-01	1.819176-01
16	2952	2.288492+01	2.308920+01	9.587209-01	2.732599-01	-1.755652-01	6.160747-01	1.899666-01
17	3198	2.399850+01	2.427604+01	0.000000	2.537582-01	0.000000	0.000000	0.000000

CASE 3
DX= 1.0000+02 METERS
RMAX= 3.542500+01 M
GRAD1=-2.1171-02(CHANNEL WIDTH GRADIENT)
GRAD2= 0.0000 HZ (AXIAL SPEED GRADIENT IN RANGE)
CO= 1.0000+00 M/SEC
A0= 1.0000+00 M
Z001= 3.6147-01 RADIANS OR 1.9985+01 DEGREES
Z0= 0.0000 M
PROFILE: 1 (1=HIRSCH,2=PARABOLIC,3=RILINEAR)

RESULTS:

MESH= 2 AT STEP 2302

MESH= 3 AT STEP 4362

MAXIMUM DISTANCE AT STEP 5209

CHANNEL WIDTH LESS THAN 0.01 AT STEP 5209

AXIS CROSSINGS: 25

SUMMARY OF AXIS CROSSINGS:

NUMBER	STEP NO.	RANGE (M)	TIME (SEC)	AVG(DR/DT) (M/SEC)	A/A0	MAX Z (M)	RP/A0	J (SEC)
1	1	0.000000	0.000000	9.979919-01	1.000000+00	3.365039-01	2.856259+00	5.523896-01
2	297	2.856259+00	2.862194+00	9.974502-01	9.995303-01	-3.261929-01	2.672060+00	5.521909-01
3	554	5.528314+00	5.540999+00	9.973149-01	8.829601-01	3.162936-01	2.499693+00	5.509326-01
4	804	8.028008+00	8.049917+00	9.969361-01	8.300391-01	-3.066939-01	2.338388+00	5.529961-01
5	1038	1.034639+01	1.039249+01	9.966027-01	7.805331-01	2.973811-01	2.187939+00	5.518509-01
6	1257	1.256383+01	1.258760+01	9.960055-01	7.347229-01	-2.884993-01	2.046160+00	5.520955-01
7	1442	1.459979+01	1.469194+01	9.954392-01	6.909037-01	2.798360-01	1.913942+00	5.490995-01
8	1653	1.651393+01	1.654967+01	9.947958-01	6.503836-01	-2.715329-01	1.790195+00	5.533706-01
9	1832	1.830512+01	1.836473+01	9.940542-01	6.124839-01	2.638299-01	1.674370+00	5.016187-01
10	1999	1.997879+01	2.004862+01	9.932111-01	5.770353-01	-2.558178-01	1.565957+00	3.672192-01
11	2154	2.154995+01	2.162529+01	9.922968-01	5.438829-01	2.483761-01	1.464973+00	3.666186-01
12	2302	2.300092+01	2.310120+01	9.913814-01	5.124781-01	-2.412829-01	1.369723+00	3.676180-01
13	2575	2.437365+01	2.447917+01	9.898807-01	4.839855-01	2.343939-01	1.280768+00	3.673859-01
14	2831	2.565491+01	2.577301+01	9.884490-01	4.564709-01	-2.277465-01	1.197481+00	3.693153-01
15	3071	2.685190+01	2.698951+01	9.868151-01	4.315185-01	2.213876-01	1.119994+00	3.656799-01
16	3296	2.797139+01	2.811896+01	9.849545-01	4.078177-01	-2.152551-01	1.046961+00	3.675425-01
17	3504	2.901785+01	2.918141+01	9.829325-01	3.854431-01	2.093588-01	9.780591-01	2.476690-01
18	3700	2.999591+01	3.017655+01	9.809129-01	3.649566-01	-2.036980-01	9.139860-01	1.840497-01
19	3883	3.090990+01	3.110880+01	9.774400-01	3.456066-01	1.982544-01	8.539577-01	1.819099-01
20	4053	3.176385+01	3.198227+01	9.795109-01	3.275275-01	-1.930389-01	7.976823-01	1.846395-01
21	4213	3.256159+01	3.280081+01	9.709255-01	3.104397-01	1.880383-01	7.449570-01	1.836303-01
22	4362	3.330699+01	3.356808+01	9.655879-01	2.944683-01	-1.832751-01	6.931019-01	1.832884-01
23	4639	3.399959+01	3.428588+01	9.621907-01	2.801996-01	1.786803-01	6.492610-01	1.825662-01
24	4892	3.466885+01	3.496065+01	9.569399-01	2.644491-01	-1.742833-01	6.057949-01	1.845954-01
25	5191	3.525465+01	3.559371+01	0.000000	2.534238-01	0.000000	0.000000	0.000000

CASE 4
DX= 1.0000-02 METERS
ZMAX= 4.723900+01 M
GRAD1=-1.5480-02(CHANNEL WIDTH GRADIENT)
GRAD2= 0.0000 MZ (AXIAL SPEED GRADIENT IN RANGE)
CO= 1.0000.00 M/SEC
AO= 1.0000.00 M
ZOOT= 3.6197-01 RADIANS OR 2.0000-01 DEGREES
ZO= 0.0000 M
PROFILE: 1 (1=HIRSCH,2=PARABOLIC,3=BILINEAR)

RESULTS:

MESH= 2 AT STEP 2973

MESH= 3 AT STEP 5910

MAXIMUM DISTANCE AT STEP 7093

CHANNEL WIDTH LESS THAN 0.01 AT STEP 7093

AXIS CROSSINGS: 32

SUMMARY OF AXIS CROSSINGS:

NUMBER	STEP NO.	RANGE (M)	TIME (SEC)	AVG(DR/DT) (M/SEC)	A/AO	MAX Z (M)	EP/AO	J (SEC)
1	1	0.000000	0.000000	9.979896-01	1.000000+00	3.380538-01	2.879600+00	7.361992-01
2	299	2.879600+00	2.885958+00	9.977579-01	9.593720-01	-3.302968-01	2.739168+00	7.360698-01
3	583	5.618768+00	5.630782+00	9.975240-01	9.187740-01	3.226937-01	2.605556+00	7.353405-01
4	874	8.229329+00	8.242805+00	9.972466-01	8.693977-01	-3.152907-01	2.478831+00	7.365326-01
5	1172	1.070276+01	1.072803+01	9.969809-01	8.300902-01	3.080383-01	2.357977+00	7.360657-01
6	1378	1.304023+01	1.309265+01	9.966639-01	7.926035-01	-3.010753-01	2.242391+00	7.354789-01
7	1532	1.530262+01	1.533954+01	9.963166-01	7.569999-01	2.991982-01	2.132088+00	7.324075-01
8	1795	1.793551+01	1.798332+01	9.959320-01	7.231241-01	-2.875557-01	2.028699+00	7.374078-01
9	1998	1.996421+01	1.952030+01	9.955057-01	6.909089-01	2.810896-01	1.929551+00	5.855151-01
10	2191	2.133776+01	2.195856+01	9.950340-01	6.602671-01	-2.748034-01	1.835212+00	5.499599-01
11	2324	2.322897+01	2.330293+01	9.945136-01	6.311240-01	2.686854-01	1.745992+00	5.513656-01
12	2498	2.497491+01	2.505800+01	9.939388-01	6.034069-01	-2.627338-01	1.660017+00	5.514275-01
13	2665	2.663493+01	2.672816+01	9.933032-01	5.770953-01	2.569975-01	1.578724+00	5.515057-01
14	2823	2.821315+01	2.831751+01	9.926016-01	5.519752-01	-2.513203-01	1.501362+00	5.520869-01
15	2973	2.971451+01	2.983006+01	9.918096-01	5.281335-01	2.458863-01	1.427920+00	5.487552-01
16	3257	3.113793+01	3.126575+01	9.909540-01	5.055376-01	-2.405638-01	1.357819+00	5.519506-01
17	3429	3.249575+01	3.263596+01	9.899890-01	4.839759-01	2.353968-01	1.291086+00	5.311752-01
18	3587	3.378634+01	3.399016+01	9.889507-01	4.634730-01	-2.303759-01	1.227609+00	3.670572-01
19	9032	3.501395+01	3.518193+01	9.877999-01	4.439785-01	2.254950-01	1.167188+00	3.665807-01
20	9266	3.618113+01	3.634303+01	9.865307-01	4.254934-01	-2.207570-01	1.109672+00	3.685699-01
21	9498	3.729081+01	3.748706+01	9.851343-01	4.078220-01	2.161560-01	1.054918+00	3.674805-01
22	9695	3.834572+01	3.855869+01	9.836450-01	3.910699-01	-2.116899-01	1.002790+00	3.660637-01
23	9898	3.934851+01	3.957827+01	9.818886-01	3.751956-01	2.073557-01	9.531589-01	3.670640-01
24	5090	4.030167+01	4.059896+01	9.800023-01	3.600099-01	-2.031997-01	9.058990-01	3.680295-01
25	5271	4.120757+01	4.147339+01	9.779337-01	3.456238-01	1.990648-01	8.608947-01	2.632086-01
26	5493	4.206897+01	4.235366+01	9.756961+01	3.319528-01	-1.951063-01	8.180332-01	1.846756-01
27	5607	4.288850+01	4.319211+01	9.731296+01	3.189629-01	1.912799-01	7.772083-01	1.827993-01
28	5762	4.366371+01	4.399078+01	9.703523-01	3.066203-01	-1.876546-01	7.383175-01	1.846197-01
29	5910	4.440201+01	4.475166+01	9.660331-01	2.948958-01	1.839908-01	6.987948-01	1.831039-01
30	6199	4.510082+01	4.547507+01	9.638877-01	2.837990-01	-1.805046-01	6.659884-01	1.846182-01
31	6456	4.576481+01	4.616596+01	9.601736-01	2.732231-01	1.771399-01	6.329909-01	1.839003-01
32	6709	4.639915+01	4.682953+01	9.561021-01	2.631815-01	-1.738697-01	6.002660-01	1.836428-01
33	6999	4.699997+01	4.745234+01	0.000000	2.536493-01	0.000000	0.000000	0.000000

```
C     THESE ROUTINES TREAT ARBITRARY DATA IN USER PROVIDED SOUND VELOCITY
C     PROFILES (SVPS). THE GIVEN SVP IS SMOOTHED USING A CUBIC SPLINE FIT
C     SUBROUTINE. RAYS ARE TRACED USING A FOURTH ORDER RUNGE-KUTTA ROUTINE
C     TO INTEGRATE THE EQUATIONS OF MOTION. FOR EACH RAY PATH WE COMPUTE THE
C     ACTION, THE AVERAGE HORIZONTAL SPEED, AND THE VERTEX VELOCITY.
      REAL R1,R2,R3,R4
      REAL J(20)
      DIMENSION R(2000), Z(2000),DR12(2000),T(2000),RTP(20),
     *ZTP(20),TTP(20),AVGSPD(20)
      DIMENSION D(200),R(2),V(5),MSG(2)
      COMMON/DATA/NPTS,ZSVP(50),CSVP(50),D,R3,J0,RAYANG,COEF(200)
C
C *** GET AND SMOOTH THE SVP DATA ***
      READ(5,100)NOSVPS
      WRITE(6,110)NOSVPS
  110 FORMAT(1H0,' ***THIS RUN CONSISTS OF',I3,' SPV(S)***')
      MSG(1)=6H
      MSG(2)=6H
      DO 90 ISVP=1,NOSVPS
      READ(5,105) NPTS,IUNITS,MSG(1),MSG(2)
  105 FORMAT(2I10,2A6)
  100 FORMAT( )
```

```
00153    23.          READ(5,100)(ZSVP(I),CSVP(I),I=1,NPTS)                                    000041
00162    24.          WRITE(6,104)MS6(1),MS6(2),IUNITS                                         000055
00167    25.      104 FORMAT(1H0/10B,'***SVP:',2A4,' ***'/                                     000065
00167    26.         * ' USER INPUT UNITS FOR THE SVP:',                                       000065
00167    27.         *  I3,' (1:M,M/SEC;2:FT,FT/SEC)')                                        000065
00150    28.          IF (IUNITS.EQ.1)WRITE(6,107)                                             000065
00153    29.          IF (IUNITS.EQ.2)WRITE(6,108)                                             000075
00156    30.      107 FORMAT(1H0/' DEPTH(M)',48,'C(M/SEC)'/)                                   000110
00157    31.      108 FORMAT(1H0/' DEPTH(FT)',48,'C(FT/SEC)'/)                                 000110
00160    32.          DO 1 I=1,NPTS                                                            000110
00163    33.        1 WRITE(6,100)ZSVP(I),CSVP(I)                                            000110
00163    34.      C ***SKIP THE PLOTTING FOR NOW***                                          000110
00170    35.          IF (IUNITS.NE.100)GO TO 5                                                000117
00172    36.          IF(IUNITS.EQ.1)                                                          000122
00172    37.         *CALL BLABEL('SOUND SPEED (M/SEC)$',100)                                  000122
00174    38.          IF(IUNITS.EQ.2)                                                          000131
00174    39.         *CALL BLABEL('SOUND SPEED (FT/SEC)$',100)                                 000131
00176    40.          IF(IUNITS.EQ.1)                                                          000140
00176    41.         *CALL YLABEL('DEPTH (M)$',100)                                            000140
00200    42.          IF(IUNITS.EQ.2)                                                          000147
00200    43.         *CALL YLABEL('DEPTH (FT)$',100)                                           000147
00202    44.          CALL LINPLT(CSVP,ZSVP,50)                                                000156
00203    45.        5 CONTINUE                                                                 000164
00203    46.      C                                                                            000164
00203    47.      C ***CONVERT TO METRIC UNITS FOR REMAINDER OF CALCULATIONS ***               000164
00204    48.          IF (IUNITS.EQ.1)GO TO 10                                                 000165
00206    49.          DO 8 I=1,NPTS                                                            000173
00211    50.          ZSVP(I)=ZSVP(I)*0.3048006                                                000173
00212    51.        8 CSVP(I)=CSVP(I)*0.3048006                                              000175
00214    52.       10 CONTINUE                                                                 000202
00214    53.      C ***FIT A SPLINE CURVE TO THE INPUT SVP***                                  000202
00215    54.                B(1)=0.                                                            000202
00216    55.                B(2)=0.                                                            000202
00217    56.                CALL SPLN1(NPTS,ZSVP,CSVP,2,0,COEF,B)                              000203
00217    57.      C *** GET INPUT DATA AND BEGIN THE RAY ANGLE LOOP ***                        000203
00220    58.                READ(5,100)DR,ZO,ZMAX,IPLOT                                        000214
00226    59.                WRITE(6,109)DR,ZO,ZMAX                                             000225
00233    60.          109   FORMAT(1H0,' INPUT PARAMETERS:'/' DR=',1PE8.2,' METERS'/           000244
00233    61.             *         ' INITIAL DEPTH=',1PE8.2,' METERS'/                         000244
00233    62.             *         ' MAX ALLOWED DEPTH=',1PE8.2,' METERS'/)                    000244
00234    63.                DO 60 IRAY=1,40                                                    000244
00237    64.                RAYANG=FLOAT(IRAY)/2.                                              000244
00240    65.                II=IRAY                                                            000251
00241    66.                V(1)=ZO                                                            000253
00242    67.                CALL SPLN2(NPTS,ZSVP,CSVP,COEF,V)                                  000255
00243    68.                CCRIT=V(2)/COS(0.01745329*RAYANG)                                  000264
00244    69.                WRITE(6,101)II,RAYANG,V(2),CCRIT                                   000275
00252    70.          101 FORMAT(1H0/' CASE NUMBER:',I4/                                       000306
00252    71.             *     ' INITIAL ANGLE=',1PE8.2,' DEGREES'/                          000306
00252    72.             *     ' INITIAL SOUND SPEED=',4PE10.1,' M/SEC'/                     000306
00252    73.             *     ' PREDICTED VERTEX SPEED=',4PE10.1,' M/SEC')                  000306
00252    74.      C                                                                          000306
00252    75.      C *** CALCULATE RAY PATH, AVG. SPEED, AND ACTION ***                       000306
00253    76.                Z(1)=ZO                                                          000306
00254    77.                R(1)=0.                                                          000310
00255    78.                DERIZ(1)=TAN(0.01745329*RAYANG)                                  000311
00256    79.                T(1)=0.                                                          000320
```

```
00257        RTP(1)=0.     $RTP IS THE TURNING POINT RANGE (ARB. INIT. VAL.)   000321
00260        TTP(1)=0.     $TTP IS THE TURNING POINT TIME  (ARB. INIT. VAL.)   000322
00261        ZTP(1)=ZO     $ZTP IS THE TURNING POINT DEPTH (ARB. INIT. VAL.)   000323
00262        ICYCLE=1      $ICYCLE COUNTS THE NUMBER OF TURNING POINTS         000325
00263        DO 12 ITEMP=1,20                                                  000332
00266     12 J(ITEMP)=0                                                        000332
00270        DO 40 I=1,2000                                                    000334
00273           IF (ICYCLE.GT.5) GO TO 45                                      000334
00275           R(I)=(I-1)*DR                                                  000342
00276           Y=Z(I)          $Y, YP, AND U ARE JUST SHORTHAND               000352
00277           YP=DZDR(I)      $FOR Z(I), DZ/DR, AND R.                       000354
00300           U=R(I)                                                         000356
00300  C                                                                       000356
00301           K1=DR*F(U,Y,YP)                                                000357
00302           K2=DR*F(U+DR/2,Y+DR*YP/2+DR*K1/8 ,YP+K1/2)                     000366
00303           K3=DR*F(U+DR/2,Y+DR*YP/2+DR*K2/8,YP+K2/2)                      000415
00304           K4=DR*F(U+DR,Y+DR*YP+DR*K3/2,YP+K3)                            000444
00305           IP1=I+1                                                        000470
00306           Z(IP1)=Z(I)+DR*(YP+(K1+K2+K3)/6.)                              000474
00307           DZDR(IP1)=YP+(K1+2*K2+2*K3+K4)/6.                              000505
00307  C                                                                       000505
00310           V(1)=0.5*(Z(IP1)+Z(I))    $V(1) IS THE AVG DEPTH ON THIS STEP  000517
00311           CALL SPLN2(NPTS,ZSVP,CSVP,COEF,V)                              000523
00312           C=V(2)                                                         000533
00313           T(IP1)=SQRT(DR*DR+(Z(IP1)-Z(I))**2)/C +T(I)                    000535
00314           J(ICYCLE)=J(ICYCLE)+(1/C)*(Z(IP1)-Z(I))*                       000551
00314        *        SIN(0.5*(ATAN(DZDR(IP1))+ATAN(DZDR(I))))                 000551
00314  C                                                                       000551
00314  C *** CHECK THE SIGN OF DZ/DR=DZDR TO SEE IF THE RAY HAS TURNED. IF IT  000551
00314  C HAS, WE INCREMENT ICYCLE, STORE THE COORDINATES OF THE TURNING POINT,  000551
00314  C AND CALCULATE THE AVERAGE HORIZONTAL SPEED (IF ICYCLE GT 1).           000551
00315           IF (DZDR(IP1).EQ.0.) GO TO 20                                  000602
00317           IF (DZDR(I)/DZDR(IP1).GT.0.) GO TO 20                          000604
00321           G=DZDR(I)/(DZDR(I)-DZDR(IP1))                                  000610
00322           RTP(ICYCLE)=R(I)+G*DR                                          000615
00323           TTP(ICYCLE)=T(I)+G*(T(IP1)-T(I))                               000620
00324           ZTP(ICYCLE)=Y+G*(Z(IP1)-Z(I))                                  000625
00325           IF (ICYCLE.EQ.1)GO TO 15                                       000632
00327           K=ICYCLE-1                                                     000635
00330           AVGSPD(K)=(RTP(ICYCLE)-RTP(K))/(TTP(ICYCLE)-TTP(K))            000642
00331     15    ICYCLE=ICYCLE+1                                                000651
00332     20    CONTINUE                                                       000654
00332  C *** CHECK TO SEE IF A SURFACE REFLECTION HAS OCCURRED. IF IT HAS, RE-  000654
00332  C VERSE THE SIGN OF Z AND DZ/DR AND USE THESE FOR THE NEW VALUES, INCRE- 000654
00332  C MENT ICYCLE, STORE THE COORDINATES OF THE TURNING POINT, AND CALCULATE 000654
00332  C THE AVERAGE SPEED.                                                     000654
00333           IF (Z(IP1).GT.0)GO TO 36                                       000655
00335           Z(IP1)=-Z(IP1)                                                 000660
00336           DZDR(IP1)=-DZDR(IP1)                                           000662
00337           G=Z(I)/(Z(I)+Z(IP1))                                           000664
00340           RTP(ICYCLE)=R(I)+G*DR                                          000670
00341           TTP(ICYCLE)=T(I)+G*(T(IP1)-T(I))                               000673
00342           ZTP(ICYCLE)=Y-G*(Z(IP1)+Z(I))                                  000700
00343           IF (ICYCLE.EQ.1)GO TO 35                                       000705
00345           K=ICYCLE-1                                                     000710
00346           AVGSPD(K)=(RTP(ICYCLE)-RTP(K))/(TTP(ICYCLE)-TTP(K))            000715
00347     35    ICYCLE=ICYCLE+1                                                000724
```

```
C ••• CHECK THE NEW DEPTH COORDINATE TO SEE IF HAS                                  000724
C ••• DEPTH HAS BEEN REACHED.  IF SO, PRINT A MESSAGE TO                           000724
C •••THIS EFFECT AND SET NEXT SVP SET.                                             000724
   30       CONTINUE                                                               000727
            IF (Z(IP1).LT.ZMAX)GO TO 40                                            000727
            WRITE(6,104)ZMAX                                                       000733
  104       FORMAT(1HO,'MAX DEPTH EXCEEDS DEPTH LIMIT:',1PE12.6,                   000741
     •      ' METERS')                                                             000741
            GO TO 90                                                               000741
   40       CONTINUE                                                               000745
C                                                                                  000745
C ••• WRITE A SUMMARY OF THE TRAJECTORY •••                                        000745
   45       WRITE(6,102)                                                           000745
            NCYCLE=ICYCLE-1                                                        000751
            DO 50 K=1,NCYCLE                                                       000754
                 V(1)=ZIP(K)                                                       000760
                 CALL SPLN2(NPTS,ZSVP,CSVP,CGRF,V)                                 000762
                 CIP=V(2)                                                          000771
   50            WRITE(6,103)K,RIP(K),TIP(K),ZIP(K),CIP,J(K),AVGSPD(K)             000773
C                                                                                  000773
C •••PLOT TRAJECTORIES UNLESS IPLOT = 0•••                                         000773
            IF (IPLOT.EQ.0)GO TO 80                                                001C11
            CALL PLABEL('RANGE (M)$',100)                                          001C13
            CALL PLABEL('DEPTH (M)$',100)                                          001C17
            CALL LINPLT(0,2,1)                                                     001C23
   80 CONTINUE                                                                     001C34
   90 CONTINUE                                                                     001C34
  102 FORMAT(1HO,' SUMMARY OF TRAJECTORY:'//'  K',6X,'RIP(K)',6X,                  001C34
     •    'TIP(K)',6X,'ZIP(K)',6X,'CIP',9X,'J(K)',6X,'AVGSP',/                    001C34
     •    1X,' (M)',8X,'(SEC)',7X,' (M)',7X,'(M/SEC)',6X,                         001C34
     •    '(SEC)',6X,'(M/SEC)')                                                   001C34
  103 FORMAT(I4,6(1PE12.6))                                                        001C34
            STOP                                                                   001C34
            END                                                                    001C40
```

```
00101        FUNCTION F(U,Y,YP)                                                    000000
00101  C THIS FUNCTION SUBROUTINE SIMPLY RETURNS THE RHS OF THE EQUATION OF       000000
00101  C MOTION OF A RAY (ASSUMING NEGLIGIBLE HORIZONTAL GRADIENTS)                000000
00101  C       D(DZ/DR)/DR=-(1+(DZ/DR)**2)*(1/C)DC/DZ                             000000
00101  C WHERE U IS THE RANGE (UNIMPORTANT HERE), Y THE DEPTH, AND YP=DZ/DR.      000000
00103        COMMON/DATA/NPTS,ZSVP(50),CSVP(50),R,DR,ZO,RAYANG,COEF(200)           000000
00104        DIMENSION V(3)                                                        000000
00105        V(1)=Y                                                                000000
00106        CALL SPLN2(NPTS,ZSVP,CSVP,COEF,V)                                     000001
00107        F=-(1+YP**2)*(1/V(2))*V(3)                                            000010
00110        RETURN                                                                000020
00111        END                                                                   000034
```

```
.RUN DAY2.ABS

 ***THIS SUB CONSISTS OF  1 SUB(S)***

          ***SUP: BASIC ATS    ***
USER INPUT UNITS FOR THE SVP:  2 (1:M,M/SEC;2:FT,FT/SEC)

DEPTH(FT)       C(FT/SEC)

  0.00000        4956.0
   200.00        4958.0
   250.00        4958.5
   300.00        4955.0
   500.00        4965.0
   750.00        4933.0
   1000.0        4920.0
   1500.0        4890.0
   1750.0        4877.0
   2000.0        4866.0
   2250.0        4859.0
   2500.0        4855.0
   2750.0        4854.0
   3000.0        4855.5
   4000.0        4864.0
   5000.0        4876.0
   7000.0        4898.0
   10000.        4947.0

 INPUT PARAMETERS:
RB= 1.00+02 METERS
INITIAL DEPTH= 1.00+02 METERS
PAR ALLOWED DEPTH= 2.00+04 METERS


CASE NUMBER:   1
INITIAL ANGLE= 5.00-01 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PROJECTED VERTEX SPEED= 1509.7+00 M/SEC

 SUMMARY OF TRAJECTORY:
```

R	RTP(R) (M)	TTP(R) (SEC)	ZTP(R) (M)	CTP (M/SEC)	J(R) (SEC)	AVGSP (M/SEC)
1	2.686582E+04	1.809187E+01	3.153575E+03	1.509699E+03	2.963917E-01	1.485139E+03
2	5.392142E+04	3.630962E+01	9.920532E+01	1.509697E+03	2.963955E-01	1.485141E+03
3	8.097702E+04	5.452695E+01	3.153588E+03	1.509699E+03	2.963967E-01	1.485143E+03
4	1.080324E+05	7.274446E+01	9.921595E+01	1.509696E+03	2.963973E-01	0.000000

CASE NUMBER: 2
INITIAL ANGLE= 1.00E+00 DEGREES
INITIAL SOUND SPEED= 1509.6E+00 M/SEC
PREDICTED VERTEX SPEED= 1509.6E+00 M/SEC

SUMMARY OF TRAJECTORY:

R	RTP(R) (M)	TTP(R) (SEC)	ZTP(R) (M)	CTP (M/SEC)	J(R) (SEC)	AVGSP (M/SEC)
1	2.668372E+04	1.797126E+01	3.163432E+03	1.509872E+03	2.984156E-01	1.485134E+03
2	5.373410E+04	3.618535E+01	9.681870E+01	1.509868E+03	2.984402E-01	1.485136E+03
3	8.078443E+04	5.439939E+01	3.163429E+03	1.509872E+03	2.984396E-01	1.485139E+03
4	1.078348E+05	7.261343E+01	9.681995E+01	1.509868E+03	2.984396E-01	0.000000

CASE NUMBER: 3
INITIAL ANGLE= 1.50E+00 DEGREES
INITIAL SOUND SPEED= 1509.6E+00 M/SEC
PREDICTED VERTEX SPEED= 1510.2E+00 M/SEC

SUMMARY OF TRAJECTORY:

R	RTP(R) (M)	TTP(R) (SEC)	ZTP(R) (M)	CTP (M/SEC)	J(R) (SEC)	AVGSP (M/SEC)
1	2.652187E+04	1.786407E+01	3.179881E+03	1.510159E+03	3.017698E-01	1.485127E+03
2	5.356446E+04	3.607300E+01	9.367384E+01	1.510156E+03	3.018507E-01	1.485129E+03
3	8.060705E+04	5.428192E+01	3.179891E+03	1.510159E+03	3.018515E-01	1.485131E+03
4	1.076496E+05	7.249086E+01	9.367609E+01	1.510156E+03	3.018516E-01	0.000000

CASE NUMBER: 4
INITIAL ANGLE= 2.00E+00 DEGREES
INITIAL SOUND SPEED= 1509.6E+00 M/SEC
PREDICTED VERTEX SPEED= 1510.6E+00 M/SEC

SUMMARY OF TRAJECTORY:

R	RTP(R) (M)	TTP(R) (SEC)	ZTP(R) (M)	CTP (M/SEC)	J(R) (SEC)	AVGSP (M/SEC)
1	2.638802E+04	1.777545E+01	3.202960E+03	1.510562E+03	3.064398E-01	1.485128E+03
2	5.343107E+04	3.598469E+01	8.823434E+01	1.510557E+03	3.066258E-01	1.485129E+03
3	8.047418E+04	5.419395E+01	3.202978E+03	1.510562E+03	3.066294E-01	1.485132E+03
4	1.075173E+05	7.240316E+01	8.821079E+01	1.510559E+03	3.066305E-01	0.000000

CASE NUMBER: 5
INITIAL ANGLE= 2.50E+00 DEGREES
INITIAL SOUND SPEED= 1509.6E+00 M/SEC

PREDICTED VERTEX SPEED= 1511.1+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.628764+04	1.770901+01	3.232784+03	1.511080+03	3.126176-01	1.485228+03
2	5.343851+04	3.598061+01	8.162665+01	1.511077+03	3.127727-01	1.485230+03
3	8.058948+04	5.427026+01	3.232732+03	1.511081+03	3.127774-01	1.485234+03
4	1.077606+05	7.255095+01	8.162031+01	1.511077+03	3.127785-01	0.000000

CASE NUMBER: 6
INITIAL ANGLE= 3.00+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1511.7+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.622690+04	1.768749+01	3.249178+03	1.511714+03	3.196974-01	1.487051+03
2	5.556652+04	3.739891+01	2.980232-08	1.510592+03	3.219740-01	1.487046+03
3	8.489949+04	5.712658+01	3.249401+03	1.511717+03	3.219550-01	1.487048+03
4	1.142312+05	7.684938+01	1.490116-08	1.510592+03	3.220563-01	0.000000

CASE NUMBER: 7
INITIAL ANGLE= 3.50+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1512.5+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.620363+04	1.765342+01	3.312534+03	1.512443+03	3.282886-01	1.486218+03
2	5.458273+04	3.674827+01	5.960464-08	1.510592+03	3.313973-01	1.486219+03
3	8.296007+04	5.584192+01	3.312690+03	1.512466+03	3.313765-01	1.486221+03
4	1.113371+05	7.493530+01	1.490116-08	1.510592+03	3.315089-01	0.000000

CASE NUMBER: 8
INITIAL ANGLE= 4.00+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1513.3+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.622782+04	1.766942+01	3.362890+03	1.513328+03	3.381973-01	1.485837+03
2	5.419878+04	3.649464+01	5.960464-08	1.510592+03	3.420140-01	1.485839+03
3	8.216946+04	5.531965+01	3.362969+03	1.513330+03	3.420237-01	1.485842+03
4	1.101406+05	7.414443+01	2.980232-08	1.510592+03	3.421698-01	0.000000

CASE NUMBER: 9
INITIAL ANGLE= 4.50+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1514.3+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.638248+04	1.771876+01	3.428529+03	1.514310+03	3.496449-01	1.485644+03
2	5.407807+04	3.641473+01	5.960464-08	1.510592+03	3.540134-01	1.485646+03
3	8.185333+04	5.511047+01	3.428869+03	1.514316+03	3.539419-01	1.485648+03
4	1.094283+05	7.380598+01	2.980232-08	1.510592+03	3.542481-01	0.000000

CASE NUMBER: 10
INITIAL ANGLE= 5.00+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1515.4+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.643431+04	1.780576+01	3.485805+03	1.515408+03	3.420683-01	1.485584+03
2	5.415141+04	3.646313+01	5.960464-08	1.510592+03	3.474279-01	1.485586+03
3	8.186872+04	5.512062+01	3.486517+03	1.515420+03	3.471605-01	1.485589+03
4	1.095860+05	7.377804+01	5.960464-08	1.510592+03	3.474248-01	0.000000

CASE NUMBER: 11
INITIAL ANGLE= 5.50+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1516.6+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.643200+04	1.793613+01	3.550167+03	1.516624+03	3.760954-01	1.485643+03
2	5.440574+04	3.662956+01	7.450581-09	1.510592+03	3.823969-01	1.485648+03
3	8.217827+04	5.532480+01	3.560908+03	1.516652+03	3.817219-01	1.485650+03
4	1.099526+05	7.401988+01	5.960464-08	1.510592+03	3.824834-01	0.000000

CASE NUMBER: 12
INITIAL ANGLE= 6.00+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1518.6+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(N) (M)	TTP(N) (SEC)	ZTP(N) (M)	CTP (M/SEC)	A(N) (SEC)	AVGSP (M/SEC)
1	2.690877+04	1.811856+01	3.643416+03	1.517957+03	3.935997-01	1.485826+03
2	5.484507+04	3.692042+01	5.960464-08	1.510592+03	3.983715-01	1.485832+03
3	8.278463+04	5.572441+01	3.642585+03	1.517976+03	3.978899-01	1.485834+03

```
  4 1.107235•05 7.452788•01 2.980232-08 1.510592•03 3.987426-01 0.000000

CASE NUMBER:  13
INITIAL ANGLE= 6.50•00 DEGREES
INITIAL SOUND SPEED= 1509.6•00 M/SEC
PREDICTED VERTEX SPEED= 1519.6•00 M/SEC

 SUMMARY OF TRAJECTORY:

  #      RTP(M)       TTP(S)       ZTP(M)       CTP          A(M)       AVGSP
          (M)         (SEC)         (M)        (M/SEC)      (SEC)       (M/SEC)
  1 2.728314•04 1.836586•01 3.733490•03 1.519600•03 4.084483-01 1.486149•03
  2 5.550280•04 3.735350•01 3.725290-09 1.510592•03 4.164387-01 1.486161•03
  3 8.373116•04 5.634764•01 3.739840•03 1.519445•03 4.153763-01 1.486164•03
  4 1.119588•05 7.534124•01 5.960464-08 1.510592•03 4.164518-01 0.000000

CASE NUMBER:  14
INITIAL ANGLE= 7.00•00 DEGREES
INITIAL SOUND SPEED= 1509.6•00 M/SEC
PREDICTED VERTEX SPEED= 1521.6•00 M/SEC

 SUMMARY OF TRAJECTORY:

  #      RTP(M)       TTP(S)       ZTP(M)       CTP          A(M)       AVGSP
          (M)         (SEC)         (M)        (M/SEC)      (SEC)       (M/SEC)
  1 2.778430•04 1.869473•01 3.836912•03 1.520979•03 4.273439-01 1.486645•03
  2 5.442957•04 3.796313•01 5.960464-08 1.510592•03 4.356244-01 1.486651•03
  3 8.507849•04 5.723301•01 3.817783•03 1.520991•03 4.345107-01 1.486655•03
  4 1.137278•05 7.650490•01 5.960464-08 1.510592•03 4.358013-01 0.000000

CASE NUMBER:  15
INITIAL ANGLE= 7.50•00 DEGREES
INITIAL SOUND SPEED= 1509.6•00 M/SEC
PREDICTED VERTEX SPEED= 1522.7•00 M/SEC

 SUMMARY OF TRAJECTORY:

  #      RTP(M)       TTP(S)       ZTP(M)       CTP          A(M)       AVGSP
          (M)         (SEC)         (M)        (M/SEC)      (SEC)       (M/SEC)
  1 2.844094•04 1.913934•01 3.934112•03 1.522668•03 4.470587-01 1.487365•03
  2 5.771847•04 3.881085•01 2.980232-08 1.510592•03 4.569604-01 1.487387•03
  3 8.699285•04 5.849188•01 3.936958•03 1.522708•03 4.557376-01 1.487391•03
  4 1.162672•05 7.817358•01 1.192093-07 1.510592•03 4.569443-01 0.000000

CASE NUMBER:  16
INITIAL ANGLE= 8.00•00 DEGREES
INITIAL SOUND SPEED= 1509.6•00 M/SEC
PREDICTED VERTEX SPEED= 1524.5•00 M/SEC

 SUMMARY OF TRAJECTORY:

  #      RTP(M)       TTP(S)       ZTP(M)       CTP          A(M)       AVGSP
          (M)         (SEC)         (M)        (M/SEC)      (SEC)       (M/SEC)
```

1	2.940026+04	1.975585+01	4.888960+03	1.524477+03	4.783791-01	1.488405+03
2	5.954228+04	4.000708+01	5.960464-08	1.510592+03	4.799268-01	1.488421+03
3	8.969892+04	6.026456+01	4.098872+03	1.524499+03	4.788174-01	1.488426+03
4	1.198515+05	8.052591+01	5.960464-08	1.510592+03	4.861819-01	0.000000

CASE NUMBER: 17
INITIAL ANGLE= 8.50+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1526.4+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(M)	TTP(S)	ZTP(M)	CTP (M/SEC)	A (S)	AVGSP (M/SEC)
1	3.078969+04	2.066665+01	4.240982+03	1.524489+03	4.953225-01	1.489969+03
2	6.227345+04	4.179740+01	1.192093-07	1.510592+03	5.051328-01	1.489964+03
3	9.376875+04	6.293570+01	4.250181+03	1.524422+03	5.045828-01	1.489969+03
4	1.252068+05	8.407389+01	1.192093-07	1.510592+03	5.054478-01	0.000000

CASE NUMBER: 18
INITIAL ANGLE= 9.00+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1528.5+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(M)	TTP(S)	ZTP(M)	CTP (M/SEC)	A (S)	AVGSP (M/SEC)
1	3.314372+04	2.220771+01	4.458972+03	1.528466+03	5.233321-01	1.492462+03
2	6.693972+04	4.485217+01	5.960464-08	1.510592+03	5.343061-01	1.492509+03
3	1.007787+05	6.752674+01	4.454099+03	1.528488+03	5.328334-01	1.492514+03
4	1.344177+05	9.019722+01	2.980232-08	1.510592+03	5.350564-01	0.000000

CASE NUMBER: 19
INITIAL ANGLE= 9.50+00 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1530.6+00 M/SEC

SUMMARY OF TRAJECTORY:

N	RTP(M)	TTP(S)	ZTP(M)	CTP (M/SEC)	A (S)	AVGSP (M/SEC)
1	3.954027+04	2.638900+01	4.744857+03	1.538633+03	5.563903-01	1.498255+03
2	7.969583+04	5.319056+01	1.192093-07	1.510592+03	5.670945-01	1.498269+03
3	1.198638+05	8.000812+01	4.765258+03	1.538633+03	5.669889-01	1.498271+03
4	1.600316+05	1.068098+02	5.960464-08	1.510592+03	5.682498-01	0.000000

CASE NUMBER: 20
INITIAL ANGLE= 1.00+01 DEGREES
INITIAL SOUND SPEED= 1509.6+00 M/SEC
PREDICTED VERTEX SPEED= 1532.9+00 M/SEC

BAY DEPTH EXCEEDS DEPTH LIMIT: 2.000000+04 METERS

REFERENCES

1. D. E. Weston, Proc. Phys. Soc. (London) *73*, 365-384 (1958).

2. R. C. Shockley, J. Acoust. Soc. Am. (to be published).

3. L. D. Landau and E. M. Lifshitz, *Mechanics*, Pergamon Press, Reading, Mass. pp 154-158 (1960).

4. D. M. Milder, J. Acoust. Soc. Am. *46*, pp 1259-1263 (1969).

5. Byrd, PF and Friedman, MD, *Handbook of Elliptic Integrals for Engineers and Scientists*, Springer-Verlag, New York (1971).

6. Mathpac Univac Corporation. SPLN1 and SPLN2 are documented in a number of manuals covering resident subroutines.

INITIAL DISTRIBUTION LIST

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
DR. G. GUSTAFSON

ARPA RESEARCH CENTER
MOFFETT FIELD, CA
R. TRUEBLOOD

NAVAL ELECTRONIC SYSTEMS COMMAND
PME-124 (CAPT H. COX)
PME-124/60 (DR. G. HETLAND)
NELEX-320 (CDR D. GRIFFITH)

NAVAL RESEARCH LABORATORY
DR. B. ADAMS
H. BROCK
DR. B. HURDLE
DR. R. FERRIS
D. NUTILE
A. GUTHRIE

NEW LONDON LABORATORY
NAVAL UNDERWATER SYSTEMS CENTER
CODE 314 (DR. H. WEINBERG)
CODE 312 (DR. F. DI NAPOLI)

THE UNIVERSITY OF TEXAS
APPLIED RESEARCH LABORATORY
P.O. BOX 8029
AUSTIN, TX 78712
DR. L. HAMPTON
DR. S. MITCHELL

BOLT BERANEK AND NEWMAN, INC.
50 MOULTON STREET
CAMBRIDGE, MA 02138
DR. P. W. SMITH

INSTITUTE FOR ACOUSTICAL RESEARCH
615 S.W. 2ND AVENUE
MIAMI, FL 33130
DR. W. JOBST
L. DOMINIJANNI

OCEAN DATA SYSTEMS, INC.
3581 KENYON STREET
SAN DIEGO, CA 92110
K. OSBORNE
DR. F. RYAN
DR. J. CROUCH

SCIENCE APPLICATIONS, INC.
8400 WESTPARK DRIVE
MC LEAN, VA 22101
C. SPOFFARD

DEFENSE DOCUMENTATION CENTER (12)